



BSI Standards Publication

Information technology — UPnP Device Architecture

Part 26-10: Telephony device control protocol —
Level 2 — Call management service

National foreword

This British Standard is the UK implementation of ISO/IEC 29341-26-10:2017.

The UK participation in its preparation was entrusted to Technical Committee ICT/-/1, Information systems co-ordination.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2017
Published by BSI Standards Limited 2017

ISBN 978 0 580 90868 2

ICS 35.200

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 September 2017.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

**Information technology — UPnP
Device Architecture —**

**Part 26-10:
Telephony device control protocol —
Level 2 — Call management service**

*Technologies de l'information — Architecture de dispositif UPnP —
Partie 26-10: Protocole de contrôle de dispositif de téléphonie —
Niveau 2 — Service de gestion des appels*





COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2017, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

CONTENTS

1	Scope.....	1
2	Normative References	1
3	Terms, definitions, symbols and abbreviated terms.....	2
4	Notations and conventions	5
4.1	Text conventions	5
4.2	Data Types.....	5
4.3	Vendor-defined Extensions	6
5	Service Modeling Definitions	6
5.1	Service Type	6
5.2	<u>CallManagement</u> Service Architecture	6
5.2.1	Managing Call With Multiple Telephony Clients	6
5.2.2	User Concept and Multi-User Control.....	10
5.2.3	Call Monopolization.....	11
5.2.4	Selective Information Delivery	16
5.2.5	Handling of Media Streams.....	16
5.2.6	Content Sharing	17
5.3	State Variables	18
5.3.1	State Variable Overview	18
5.3.2	<u>CallInfo</u>	19
5.3.3	<u>TelCPNameList</u>	23
5.3.4	<u>CallBackAvailability</u>	23
5.3.5	<u>PushInfo</u>	23
5.3.6	<u>VoiceMailInfo</u>	24
5.3.7	<u>ParallelCallInfo</u>	25
5.3.8	<u>A ARG TYPE TelephonyServerIdentity</u>	26
5.3.9	<u>A ARG TYPE TelCPName</u>	26
5.3.10	<u>A ARG TYPE TelCPNameList</u>	26
5.3.11	<u>A ARG TYPE Expires</u>	26
5.3.12	<u>A ARG TYPE MediaCapabilityInfo</u>	26
5.3.13	<u>A ARG TYPE CalleeID</u>	28
5.3.14	<u>A ARG TYPE CallPriority</u>	28
5.3.15	<u>A ARG TYPE CallMode</u>	29
5.3.16	<u>A ARG TYPE CallID</u>	29
5.3.17	<u>A ARG TYPE SecretKey</u>	29
5.3.18	<u>A ARG TYPE RejectReason</u>	29
5.3.19	<u>A ARG TYPE TCList</u>	29
5.3.20	<u>A ARG TYPE CallInfoList</u>	30
5.3.21	<u>A ARG TYPE CallLogs</u>	31
5.3.22	<u>A ARG TYPE CallBackID</u>	32
5.3.23	<u>A ARG TYPE CallBackInfoList</u>	32
5.3.24	<u>A ARG TYPE PushInfoList</u>	33
5.3.25	<u>A ARG TYPE VoiceMailInfoList</u>	34
5.3.26	<u>A ARG TYPE VoiceMailID</u>	34

ISO/IEC 29341-26-10:2017(E)

5.3.27	<u>A ARG TYPE CallType</u>	34
5.3.28	<u>A ARG TYPE CallerID</u>	35
5.3.29	<u>A ARG TYPE MaxWaitingTime</u>	36
5.4	Eventing and Moderation	36
5.4.1	Eventing of <u>CallInfo</u>	36
5.4.2	Eventing of <u>TelCPNameList</u>	37
5.4.3	Eventing of <u>CallBackAvailability</u>	37
5.4.4	Eventing of <u>PushInfo</u>	37
5.4.5	Eventing of <u>VoiceMailInfo</u>	37
5.4.6	Eventing of <u>ParallelCallInfo</u>	37
5.5	Actions	37
5.5.1	<u>GetTelephonyIdentity()</u>	38
5.5.2	<u>RegisterTelCPName()</u>	39
5.5.3	<u>UnregisterTelCPName()</u>	41
5.5.4	<u>ChangeTelCPName()</u>	41
5.5.5	<u>GetTelCPNameList()</u>	42
5.5.6	<u>GetMediaCapabilities()</u>	43
5.5.7	<u>StartCall()</u>	44
5.5.8	<u>StopCall()</u>	46
5.5.9	<u>AcceptCall()</u>	48
5.5.10	<u>RejectCall()</u>	50
5.5.11	<u>ModifyCall()</u>	52
5.5.12	<u>AcceptModifyCall()</u>	54
5.5.13	<u>StartMediaTransfer()</u>	56
5.5.14	<u>ChangeMonopolizer()</u>	57
5.5.15	<u>InitiateCall()</u>	59
5.5.16	<u>GetCallInfo()</u>	59
5.5.17	<u>GetCallLogs()</u>	61
5.5.18	<u>ClearCallLogs()</u>	61
5.5.19	<u>RegisterCallBack()</u>	62
5.5.20	<u>ClearCallBack()</u>	63
5.5.21	<u>GetCallBackInfo()</u>	63
5.5.22	<u>ChangeCallMode()</u>	64
5.5.23	<u>GetPushInfo()</u>	65
5.5.24	<u>IgnoreCall()</u>	66
5.5.25	<u>GetVoiceMail()</u>	67
5.5.26	<u>DeleteVoiceMail()</u>	68
5.5.27	<u>EnhancedInitiateCall()</u>	69
5.5.28	<u>WaitingForCall()</u>	70
5.5.29	<u>InitiateParallelCall()</u>	71
5.5.30	<u>AcceptParallelCall()</u>	72
5.5.31	Relationships Between Actions	73
5.5.32	Error Code Summary	73
5.6	Service Behavioral Model	73
5.6.1	State Diagram	73
6	XML Service Description	77
Annex A (normative)	XML complex type <i>peerType</i>	93

Annex B (normative) XML Schema	97
Annex C (informative) Theory of Operation	102
Annex D (informative) Sequence Examples	125
Annex E (informative) How to send DTMF	295
Annex F (informative) Bibliography	296
Figure 1 — Relationship among Media Stream, Media Session and Media Session ID	3
Figure 2 — Architecture of the <u>CallManagement</u> Service	6
Figure 3 — Managing a Telephony Call with Multiple Telephony Clients	7
Figure 4 — Starting a Call with Multiple Telephony Clients Using Media Mixing Capability	8
Figure 5 — Modifying a Call with Multiple Telephony Clients Using Media Mixing Capability	8
Figure 6 — Starting a Call With Multiple Telephony Clients Using Media Sharing Capability	9
Figure 7 — Modifying a Call With Multiple Telephony Clients Using Media Sharing Capability	10
Figure 8 — Example for Multi-User Control Concept	11
Figure 9 — PHONE-based Call Monopolization	13
Figure 10 — Incoming Call Handling	15
Figure 11 — Call State Diagram for a Caller	74
Figure 12 — Call State Diagram for a Callee	75
Figure 13 — Call State Diagram to request the modification of a Call	76
Figure 14 — Call State Diagram to accept the modification of a Call	77
Figure C.1 — Flow basics for creating an outgoing Call (TC-Based Media Handling)	103
Figure C.2 — Flow basics for accepting an incoming Call (TC-Based Media Handling)	105
Figure C.3 — Flow basics for modifying an ongoing Call (TC-Based Media Handling)	107
Figure C.4 — Flow basics for creating an outgoing Call (TS-Based Media Handling)	109
Figure C.5 — Flow basics for accepting an incoming Call (TS-Based Media Handling)	111
Figure C.6 — Flow basics for modifying an ongoing Call (TS-Based Media Handling)	113
Figure C.7 — Flow basics for changing CallMode from Non-Monopolize to Monopolize	114
Figure C.8 — Flow basics for changing CallMode from Monopolize to Non-Monopolize	115
Figure C.9 — Flow basics for Early Media	116
Figure C.10 — Example flow of TS-Based Content Sharing by initiating a Call	118
Figure C.11 — Example flow of TS-Based Content Sharing in an existing Call	119
Figure C.12 — Example flow of TC-Based Content Sharing by Updating a Call	120
Figure C.13 — Example flow of TC-Based Content Sharing without Updating a Call	121
Figure C.14 — Example flow of TC-Based Content Sharing by initiating a Call	122
Figure C.15 — Architecture for Parallel Call	123
Figure C.16 — Flow basics for Establishing a Parallel Call	124
Figure D.1 — Register TelCPName	125
Figure D.2 — Register TelCPName (But the specified TelCPName is already in use.)	126
Figure D.3 — Keep using the same TelCPName	127

ISO/IEC 29341-26-10:2017(E)

Figure D.4 — Keep using the same TelCPName (But the specified Secret Key is invalid.)	128
Figure D.5 — Change TelCPName	129
Figure D.6 — Change TelCPName (But the specified Secret Key is invalid.)	130
Figure D.7 — Change TelCPName (But the specified TelCPName is already in use.)	131
Figure D.8 — Unregister TelCPName	132
Figure D.9 — Unregister TelCPName (But the specified TelCPName does not exist.)	133
Figure D.10 — Get existing TelCPNames	134
Figure D.11 — Get existing TelCPNames (But no TelCP exists)	135
Figure D.12 — Create a Call	137
Figure D.13 — Terminate a Call	153
Figure D.14 — Reject an incoming Call	158
Figure D.15 — Cancel an outgoing Call	161
Figure D.16 — Cancel an outgoing Call (But the specified CallID does not exist.)	164
Figure D.17 — Cancel an outgoing Call (But the StopCall() action is invoked at invalid timing.)	165
Figure D.18 — Get MediaSessionInfo and CallInfo during the Call	166
Figure D.19 — Create an outgoing Call using multiple TCs (Case 1)	169
Figure D.20 — Create an outgoing Call using multiple TCs (Case 2)	183
Figure D.21 — Accept an incoming Call using multiple TCs (Case 1)	188
Figure D.22 — Accept an incoming Call using multiple TCs (Case 2)	201
Figure D.23 — Add TC during a Call (Create a modification request)	205
Figure D.24 — Add TC during a Call (Accept a modification request)	215
Figure D.25 — Remove TC during a Call (Create a modification request)	224
Figure D.26 — Remove TC during a Call (Accept a modification request)	230
Figure D.27 — Change TC during a Call	237
Figure D.28 — Modify Media Session during a Call	246
Figure D.29 — Initiate a Call	254
Figure D.30 — Create an outgoing Call (Monopolization Mode Call(PHONE-TelCP))	257
Figure D.31 — Terminate a Call (But the specified TelCP is not authorized.)	272
Figure D.32 — Cancel an outgoing Call (But the specified TelCP is not authorized.)	273
Figure D.33 — Change TelCP which Monopolize the Call	274
Figure D.34 — Create a Call with TC-Based and TS-Based Media Handlings	277
Table 1 — State Variables	18
Table 2 — Allowed values for type	20
Table 3 — Allowed values for callStatus	20
Table 4 — Allowed values for reason and the corresponding values of the Call Status	21
Table 5 — Allowed values for priority	22
Table 6 — Allowed values for format	22
Table 7 — Allowed values for priority	24
Table 8 — Allowed values for informationType	26

Table 9 — allowedValueList for the <i>A_ARG_TYPE_CallPriority</i> state variable	28
Table 10 — allowedValueList for the <i>A_ARG_TYPE_CallMode</i> state variable	29
Table 11 — Allowed values for videoQuality	35
Table 12 — Allowed values for audioQuality	35
Table 13 — Event Moderation	36
Table 14 — Actions	37
Table 15 — Arguments for <i>GetTelephonyIdentity()</i>	38
Table 16 — Error Codes for <i>GetTelephonyIdentity()</i>	39
Table 17 — Arguments for <i>RegisterTelCPName()</i>	39
Table 18 — Error Codes for <i>RegisterTelCPName()</i>	40
Table 19 — Arguments for <i>UnregisterTelCPName()</i>	41
Table 20 — Error Codes for <i>UnregisterTelCPName()</i>	41
Table 21 — Arguments for <i>ChangeTelCPName()</i>	41
Table 22 — Error Codes for <i>ChangeTelCPName()</i>	42
Table 23 — Arguments for <i>GetTelCPNameList()</i>	43
Table 24 — Error Codes for <i>GetTelCPNameList()</i>	43
Table 25 — Arguments for <i>GetMediaCapabilities()</i>	43
Table 26 — Error Codes for <i>GetMediaCapabilities()</i>	44
Table 27 — Arguments for <i>StartCall()</i>	44
Table 28 — Error Codes for <i>StartCall()</i>	46
Table 29 — Arguments for <i>StopCall()</i>	46
Table 30 — Error Codes for <i>StopCall()</i>	48
Table 31 — Arguments for <i>AcceptCall()</i>	48
Table 32 — Error Codes for <i>AcceptCall()</i>	50
Table 33 — Arguments for <i>RejectCall()</i>	50
Table 34 — Error Codes for <i>RejectCall()</i>	52
Table 35 — Arguments for <i>ModifyCall()</i>	52
Table 36 — Error Codes for <i>ModifyCall()</i>	53
Table 37 — Arguments for <i>AcceptModifyCall()</i>	54
Table 38 — Error Codes for <i>AcceptModifyCall()</i>	55
Table 39 — Arguments for <i>StartMediaTransfer()</i>	56
Table 40 — Error Codes for <i>StartMediaTransfer()</i>	57
Table 41 — Arguments for <i>ChangeMonopolizer()</i>	57
Table 42 — Error Codes for <i>ChangeMonopolizer()</i>	58
Table 43 — Arguments for <i>InitiateCall()</i>	59
Table 44 — Error Codes for <i>InitiateCall()</i>	59
Table 45 — Arguments for <i>GetCallInfo()</i>	59
Table 46 — Error Codes for <i>GetCallInfo()</i>	60
Table 47 — Arguments for <i>GetCallLogs()</i>	61
Table 48 — Error Codes for <i>GetCallLogs()</i>	61
Table 49 — Error Codes for <i>ClearCallLogs()</i>	62
Table 50 — Arguments for <i>RegisterCallBack()</i>	62

ISO/IEC 29341-26-10:2017(E)

Table 51 — Error Codes for <i>RegisterCallBack()</i>	63
Table 52 — Arguments for <i>ClearCallBack()</i>	63
Table 53 — Error Codes for <i>ClearCallBack()</i>	63
Table 54 — Arguments for <i>GetCallBackInfo()</i>	63
Table 55 — Error Codes for <i>GetCallBackInfo()</i>	64
Table 56 — Arguments for <i>ChangeCallMode()</i>	64
Table 57 — Error Codes for <i>ChangeCallMode()</i>	65
Table 58 — Arguments for <i>GetPushInfo()</i>	65
Table 59 — Error Codes for <i>GetPushInfo()</i>	66
Table 60 — Arguments for <i>IgnoreCall()</i>	66
Table 61 — Error Codes for <i>IgnoreCall()</i>	67
Table 62 — Arguments for <i>GetVoiceMail()</i>	67
Table 63 — Error Codes for <i>GetVoiceMail()</i>	68
Table 64 — Arguments for <i>DeleteVoiceMail()</i>	68
Table 65 — Error Codes for <i>DeleteVoiceMail()</i>	69
Table 66 — Arguments for <i>EnhancedInitiateCall()</i>	69
Table 67 — Error Codes for <i>EnhancedInitiateCall()</i>	70
Table 68 — Arguments for <i>WaitingForCall()</i>	70
Table 69 — Error Codes for <i>WaitingForCall()</i>	71
Table 70 — Arguments for <i>InitiateParallelCall()</i>	71
Table 71 — Error Codes for <i>InitiateParallelCall()</i>	72
Table 72 — Arguments for <i>AcceptParallelCall()</i>	72
Table 73 — Error Codes for <i>AcceptParallelCall()</i>	72
Table 74 — Error Code Summary	73

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see <http://www.iso.org/directives>).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of Standard, the meaning of the ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: [Foreword – Supplementary information](#)

ISO/IEC 29341-26-10 was prepared by UPnP Forum and adopted, under the PAS procedure, by joint technical committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

The list of all currently available parts of ISO/IEC 29341 series, under the general title *Information technology — UPnP Device Architecture*, can be found on the [ISO web site](#).

Introduction

ISO and IEC draw attention to the fact that it is claimed that compliance with this document may involve the use of patents as indicated below.

ISO and IEC take no position concerning the evidence, validity and scope of these patent rights. The holders of -these patent rights have assured ISO and IEC that they are willing to negotiate licenses under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statements of the holders of these patent rights are registered with ISO and IEC.

Intel Corporation has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Intel Corporation
Standards Licensing Department
5200 NE Elam Young Parkway
MS: JFS-98
USA – Hillsboro, Oregon 97124

Microsoft Corporation has informed IEC and ISO that it has patent applications or granted patents as listed below:

6101499 / US; 6687755 / US; 6910068 / US; 7130895 / US; 6725281 / US; 7089307 / US;
7069312 / US; 10/783 524 /US

Information may be obtained from:

Microsoft Corporation
One Microsoft Way
USA – Redmond WA 98052

Philips International B.V. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Philips International B.V. – IP&S
High Tech campus, building 44 3A21
NL – 5656 Eindhoven

NXP B.V. (NL) has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

NXP B.V. (NL)
High Tech campus 60
NL – 5656 AG Eindhoven

Matsushita Electric Industrial Co. Ltd. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Matsushita Electric Industrial Co. Ltd.
1-3-7 Shiromi, Chuoh-ku
JP – Osaka 540-6139

Hewlett Packard Company has informed IEC and ISO that it has patent applications or granted patents as listed below:

5 956 487 / US; 6 170 007 / US; 6 139 177 / US; 6 529 936 / US; 6 470 339 / US; 6 571 388 / US; 6 205 466 / US

Information may be obtained from:

Hewlett Packard Company
1501 Page Mill Road
USA – Palo Alto, CA 94304

Samsung Electronics Co. Ltd. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Digital Media Business, Samsung Electronics Co. Ltd.
416 Maetan-3 Dong, Yeongtang-Gu,
KR – Suwon City 443-742

Huawei Technologies Co., Ltd. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Huawei Technologies Co., Ltd.
Administration Building, Bantian Longgang District
Shenzhen – China 518129

Qualcomm Incorporated has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Qualcomm Incorporated
5775 Morehouse Drive
San Diego, CA – USA 92121

Telecom Italia S.p.A. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Telecom Italia S.p.A.
Via Reiss Romoli, 274
Turin - Italy 10148

Cisco Systems informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA – USA 95134

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 29341-26-10:2017(E)

Original UPnP Document

Reference may be made in this document to original UPnP documents. These references are retained in order to maintain consistency between the specifications as published by ISO/IEC and by UPnP Implementers Corporation and later by UPnP Forum. The following table indicates the original UPnP document titles and the corresponding part of ISO/IEC 29341:

UPnP Document Title	ISO/IEC 29341 Part
UPnP Device Architecture 1.0	ISO/IEC 29341-1:2008
UPnP Device Architecture Version 1.0	ISO/IEC 29341-1:2011
UPnP Device Architecture 1.1	ISO/IEC 29341-1-1:2011
UPnP Device Architecture 2.0	ISO/IEC 29341-1-2
UPnP Basic:1 Device	ISO/IEC 29341-2
UPnP AV Architecture:1	ISO/IEC 29341-3-1:2008
UPnP AV Architecture:1	ISO/IEC 29341-3-1:2011
UPnP AVTransport:1 Service	ISO/IEC 29341-3-10
UPnP ConnectionManager:1 Service	ISO/IEC 29341-3-11
UPnP ContentDirectory:1 Service	ISO/IEC 29341-3-12
UPnP RenderingControl:1 Service	ISO/IEC 29341-3-13
UPnP MediaRenderer:1 Device	ISO/IEC 29341-3-2
UPnP MediaRenderer:2 Device	ISO/IEC 29341-3-2:2011
UPnP MediaServer:1 Device	ISO/IEC 29341-3-3
UPnP AVTransport:2 Service	ISO/IEC 29341-4-10:2008
UPnP AVTransport:2 Service	ISO/IEC 29341-4-10:2011
UPnP ConnectionManager:2 Service	ISO/IEC 29341-4-11:2008
UPnP ConnectionManager:2 Service	ISO/IEC 29341-4-11:2011
UPnP ContentDirectory:2 Service	ISO/IEC 29341-4-12
UPnP RenderingControl:2 Service	ISO/IEC 29341-4-13:2008
UPnP RenderingControl:2 Service	ISO/IEC 29341-4-13:2011
UPnP ScheduledRecording:1	ISO/IEC 29341-4-14
UPnP ScheduledRecording:2	ISO/IEC 29341-4-14:2011
UPnP MediaRenderer:2 Device	ISO/IEC 29341-4-2
UPnP MediaServer:2 Device	ISO/IEC 29341-4-3
UPnP AV Datastructure Template:1	ISO/IEC 29341-4-4:2008
UPnP AV Datastructure Template:1	ISO/IEC 29341-4-4:2011
UPnP DigitalSecurityCamera:1 Device	ISO/IEC 29341-5-1
UPnP DigitalSecurityCameraMotionImage:1 Service	ISO/IEC 29341-5-10
UPnP DigitalSecurityCameraSettings:1 Service	ISO/IEC 29341-5-11
UPnP DigitalSecurityCameraStillImage:1 Service	ISO/IEC 29341-5-12
UPnP HVAC_System:1 Device	ISO/IEC 29341-6-1
UPnP ControlValve:1 Service	ISO/IEC 29341-6-10
UPnP HVAC_FanOperatingMode:1 Service	ISO/IEC 29341-6-11
UPnP FanSpeed:1 Service	ISO/IEC 29341-6-12
UPnP HouseStatus:1 Service	ISO/IEC 29341-6-13
UPnP HVAC_SetpointSchedule:1 Service	ISO/IEC 29341-6-14
UPnP TemperatureSensor:1 Service	ISO/IEC 29341-6-15
UPnP TemperatureSetpoint:1 Service	ISO/IEC 29341-6-16
UPnP HVAC_UserOperatingMode:1 Service	ISO/IEC 29341-6-17

UPnP HVAC_ZoneThermostat:1 Device	ISO/IEC 29341-6-2
UPnP BinaryLight:1 Device	ISO/IEC 29341-7-1
UPnP Dimming:1 Service	ISO/IEC 29341-7-10
UPnP SwitchPower:1 Service	ISO/IEC 29341-7-11
UPnP DimmableLight:1 Device	ISO/IEC 29341-7-2
UPnP InternetGatewayDevice:1 Device	ISO/IEC 29341-8-1
UPnP LANHostConfigManagement:1 Service	ISO/IEC 29341-8-10
UPnP Layer3Forwarding:1 Service	ISO/IEC 29341-8-11
UPnP LinkAuthentication:1 Service	ISO/IEC 29341-8-12
UPnP RadiusClient:1 Service	ISO/IEC 29341-8-13
UPnP WANCableLinkConfig:1 Service	ISO/IEC 29341-8-14
UPnP WANCommonInterfaceConfig:1 Service	ISO/IEC 29341-8-15
UPnP WANDSLLinkConfig:1 Service	ISO/IEC 29341-8-16
UPnP WANEthernetLinkConfig:1 Service	ISO/IEC 29341-8-17
UPnP WANIPConnection:1 Service	ISO/IEC 29341-8-18
UPnP WANPOTSLinkConfig:1 Service	ISO/IEC 29341-8-19
UPnP LANDevice:1 Device	ISO/IEC 29341-8-2
UPnP WANPPPConnection:1 Service	ISO/IEC 29341-8-20
UPnP WLANConfiguration:1 Service	ISO/IEC 29341-8-21
UPnP WANDevice:1 Device	ISO/IEC 29341-8-3
UPnP WANConnectionDevice:1 Device	ISO/IEC 29341-8-4
UPnP WLANAccessPointDevice:1 Device	ISO/IEC 29341-8-5
UPnP Printer:1 Device	ISO/IEC 29341-9-1
UPnP ExternalActivity:1 Service	ISO/IEC 29341-9-10
UPnP Feeder:1.0 Service	ISO/IEC 29341-9-11
UPnP PrintBasic:1 Service	ISO/IEC 29341-9-12
UPnP Scan:1 Service	ISO/IEC 29341-9-13
UPnP Scanner:1.0 Device	ISO/IEC 29341-9-2
UPnP QoS Architecture:1.0	ISO/IEC 29341-10-1
UPnP QosDevice:1 Service	ISO/IEC 29341-10-10
UPnP QosManager:1 Service	ISO/IEC 29341-10-11
UPnP QosPolicyHolder:1 Service	ISO/IEC 29341-10-12
UPnP QoS Architecture:2	ISO/IEC 29341-11-1
UPnP QosDevice:2 Service	ISO/IEC 29341-11-10
UPnP QosManager:2 Service	ISO/IEC 29341-11-11
UPnP QosPolicyHolder:2 Service	ISO/IEC 29341-11-12
UPnP QOS v2 Schema Files	ISO/IEC 29341-11-2
UPnP RemoteUIClientDevice:1 Device	ISO/IEC 29341-12-1
UPnP RemoteUIClient:1 Service	ISO/IEC 29341-12-10
UPnP RemoteUIServer:1 Service	ISO/IEC 29341-12-11
UPnP RemoteUIServerDevice:1 Device	ISO/IEC 29341-12-2
UPnP DeviceSecurity:1 Service	ISO/IEC 29341-13-10
UPnP SecurityConsole:1 Service	ISO/IEC 29341-13-11
UPnP ContentDirectory:3 Service	ISO/IEC 29341-14-12:2011
UPnP MediaServer:3 Device	ISO/IEC 29341-14-3:2011
UPnP ContentSync:1	ISO/IEC 29341-15-10:2011

ISO/IEC 29341-26-10:2017(E)

UPnP Low Power Architecture:1	ISO/IEC 29341-16-1:2011
UPnP LowPowerProxy:1 Service	ISO/IEC 29341-16-10:2011
UPnP LowPowerDevice:1 Service	ISO/IEC 29341-16-11:2011
UPnP QoS Architecture:3	ISO/IEC 29341-17-1:2011
UPnP QosDevice:3 Service	ISO/IEC 29341-17-10:2011
UPnP QosManager:3 Service	ISO/IEC 29341-17-11:2011
UPnP QosPolicyHolder:3 Service	ISO/IEC 29341-17-12:2011
UPnP QosDevice:3 Addendum	ISO/IEC 29341-17-13:2011
UPnP RemoteAccessArchitecture:1	ISO/IEC 29341-18-1:2011
UPnP InboundConnectionConfig:1 Service	ISO/IEC 29341-18-10:2011
UPnP RADAConfig:1 Service	ISO/IEC 29341-18-11:2011
UPnP RADASync:1 Service	ISO/IEC 29341-18-12:2011
UPnP RATAConfig:1 Service	ISO/IEC 29341-18-13:2011
UPnP RAClient:1 Device	ISO/IEC 29341-18-2:2011
UPnP RAServer:1 Device	ISO/IEC 29341-18-3:2011
UPnP RADiscoveryAgent:1 Device	ISO/IEC 29341-18-4:2011
UPnP SolarProtectionBlind:1 Device	ISO/IEC 29341-19-1:2011
UPnP TwoWayMotionMotor:1 Service	ISO/IEC 29341-19-10:2011
UPnP AV Architecture:2	ISO/IEC 29341-20-1
UPnP AVTransport:3 Service	ISO/IEC 29341-20-10
UPnP ConnectionManager:3 Service	ISO/IEC 29341-20-11
UPnP ContentDirectory:4 Device	ISO/IEC 29341-20-12
UPnP RenderingControl:3 Service	ISO/IEC 29341-20-13
UPnP ScheduledRecording:2 Service	ISO/IEC 29341-20-14
UPnP MediaRenderer:3 Service	ISO/IEC 29341-20-2
UPnP MediaServer:4 Device	ISO/IEC 29341-20-3
UPnP AV Datastructure Template:1	ISO/IEC 29341-20-4
UPnP InternetGatewayDevice:2 Device	ISO/IEC 29341-24-1
UPnP WANIPConnection:2 Service	ISO/IEC 29341-24-10
UPnP WANIPv6FirewallControl:1 Service	ISO/IEC 29341-24-11
UPnP WANConnectionDevice:2 Service	ISO/IEC 29341-24-2
UPnP WANDevice:2 Device	ISO/IEC 29341-24-3
UPnP Telephony Architecture:2	ISO/IEC 29341-26-1
UPnP CallManagement:2 Service	ISO/IEC 29341-26-10
UPnP MediaManagement:2 Service	ISO/IEC 29341-26-11
UPnP Messaging:2 Service	ISO/IEC 29341-26-12
UPnP PhoneManagement:2 Service	ISO/IEC 29341-26-13
UPnP AddressBook:1 Service	ISO/IEC 29341-26-14
UPnP Calendar:1 Service	ISO/IEC 29341-26-15
UPnP Presense:1 Service	ISO/IEC 29341-26-16
UPnP TelephonyClient:2 Device	ISO/IEC 29341-26-2
UPnP TelephonyServer:2 Device	ISO/IEC 29341-26-3
UPnP Friendly Info Update:1 Service	ISO/IEC 29341-27-1
UPnP MultiScreen MultiScreen Architecture:1	ISO/IEC 29341-28-1
UPnP MultiScreen Application Management:1 Service	ISO/IEC 29341-28-10
UPnP MultiScreen Screen:1 Device	ISO/IEC 29341-28-2

ISO/IEC 29341-26-10:2017(E)

UPnP MultiScreen Application Management:2 Service	ISO/IEC 29341-29-10
UPnP MultiScreen Screen:2 Device	ISO/IEC 29341-29-2
UPnP IoT Management and Control Architecture Overview:1	ISO/IEC 29341-30-1
UPnP DataStore:1 Service	ISO/IEC 29341-30-10
UPnP IoT Management and Control Data Model:1 Service	ISO/IEC 29341-30-11
UPnP IoT Management and Control Transport Generic:1 Service	ISO/IEC 29341-30-12
UPnP IoT Management and Control:1 Device	ISO/IEC 29341-30-2
UPnP Energy Management:1 Service	ISO/IEC 29341-31-1

1 Scope

This service definition is compliant with [1]. It defines a service type referred to herein as the CallManagement service.

The CallManagement service is a UPnP service that allows control points to use the telephony features(e.g., voice call, video call, and data transfer etc.) provided by a Telephony Server (TS).

The CallManagement service enables the following features to a Telephony Control Point (TelCP):

- start an outgoing call
- accept an incoming call
- modify the capability of an existing call (e.g., changing from a voice call to a video call)
- terminate a call
- preview call logs
- register a call back and check the availability of the registered call back

To realize these features, the CallManagement service manages the connectivity of calls and media streams.

2 Normative References

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

[1] – UPnP Device Architecture, version 1.0, UPnP Forum, October 15, 2008. Available at: <http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.0-20081015.pdf>. Latest version available at: <http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.0.pdf>.

[2] – Data elements and interchange formats – Information interchange -- Representation of dates and times, International Standards Organization, December 21, 2000. Available at: [ISO 8601:2000](http://www.iso.org/iso/8601).

[3] – IETF RFC 2396, Uniform Resource Identifiers (URI): Generic Syntax, T. Berners-Lee, MIT/LCS, R. Fielding, U.C. Irvine, L. Masinter, Xerox Corporation, 1998. Available at: <http://www.ietf.org/rfc/rfc2396.txt>.

[4] – IETF RFC 3261, SIP: Session Initiation Protocol, J. Rosenberg, dynamicsoft, H. Schulzrinne, Columbia U., G. Camarillo, Ericsson, A. Johnston, WorldCom, J. Peterson, Neustar, R. Sparks, dynamicsoft, M. Handley, ICIR, E. Schooler, AT&T, 2002. Available at: <http://www.ietf.org/rfc/rfc3261.txt>.

[5] – IETF RFC 3339, Date and Time on the Internet: Timestamps, G. Klyne, Clearswift Corporation, C. Newman, Sun Microsystems, July 2002. Available at: <http://www.ietf.org/rfc/rfc3339.txt>.

[6] – IETF RFC 3966, The tel URI for Telephone Numbers, H. Schulzrinne, Columbia University, 2004. Available at: <http://www.ietf.org/rfc/rfc3966.txt>.

[7] – IETF RFC 4566, SDP: Session Description Protocol, M. Handley, UCL, V. Jacobson, Packet Design, C. Perkins, University of Glasgow, July 2006. Available at: <http://www.ietf.org/rfc/rfc4566.txt>.

ISO/IEC 29341-26-10:2017(E)

[8] – Extensible Markup Language (XML) 1.0 (Third Edition), François Yergeau, Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, eds., W3C Recommendation, February 4, 2004. Available at: <http://www.w3.org/TR/2004/REC-xml-20040204>.

[9] – XML Schema Part 2: Data Types, Second Edition, Paul V. Biron, Ashok Malhotra, W3C Recommendation, 28 October 2004. Available at: <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028>.

[10] – *MediaManagement:2*, UPnP Forum, December 10, 2012. Available at: <http://www.upnp.org/specs/phone/UPnP-phone-MediaManagement-v2-Service-20121210.pdf>. Latest version available at: <http://www.upnp.org/specs/phone/UPnP-phone-MediaManagement-Service.pdf>.

[11] – IETF RFC 2046, Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types, N. Freed, Innosft, N. Borenstein, First Virtual, November 1996. Available at: <http://www.ietf.org/rfc/rfc2046.txt>.

[12] – *ConnectionManager:1*, UPnP Forum, June 25, 2002. Available at: <http://www.upnp.org/specs/av/UPnP-av-ConnectionManager-v1-Service-20020625.pdf>. Latest version available at: <http://www.upnp.org/specs/av/UPnP-av-ConnectionManager-Service.pdf>.

3 Terms, definitions, symbols and abbreviated terms

For the purposes of this document, the terms and definitions given in [1] and the following apply.

3.1 Provisioning terms

3.1.1

conditionally allowed

CA

The definition or behavior depends on a condition. If the specified condition is met, then the definition or behavior is allowed, otherwise it is not allowed.

3.1.2

conditionally required

CR

The definition or behavior depends on a condition. If the specified condition is met, then the definition or behavior is required, otherwise it is not allowed.

3.1.3

not allowed

The definition or behavior is prohibited by this specification. Opposite of required.

3.2 Symbols

3.2.1

::

signifies a hierarchical parent-child (parent::child) relationship between the two objects separated by the double colon. This delimiter is used in multiple contexts, for example: Service::Action(), Action()::Argument, parentProperty::childProperty.

3.3 Terms

3.3.1

Call

connection between parties that are communicating with each other. The connection can be for voice, video, or data transmission.

3.3.2

CallID

unique identifier of a Call.

3.3.3**Call Status**

indicator of the state(e.g., Calling, Ringing, and Talking etc.) of a Call at a particular point in time.

3.3.4**Media Stream**

a flow of media (e.g., audio, video, and data etc.) which is sent and/or received between two parties involved in a Call.

3.3.5**Media Session**

series of interactions for flow of media among the parties involved in the Call.

3.3.6**Media Session ID**

unique identifier of the Media Session for a Call.

Figure 1 shows the relationship among Media Streams, Media Session and Media Session ID. A Media Session can include multiple Media Streams.

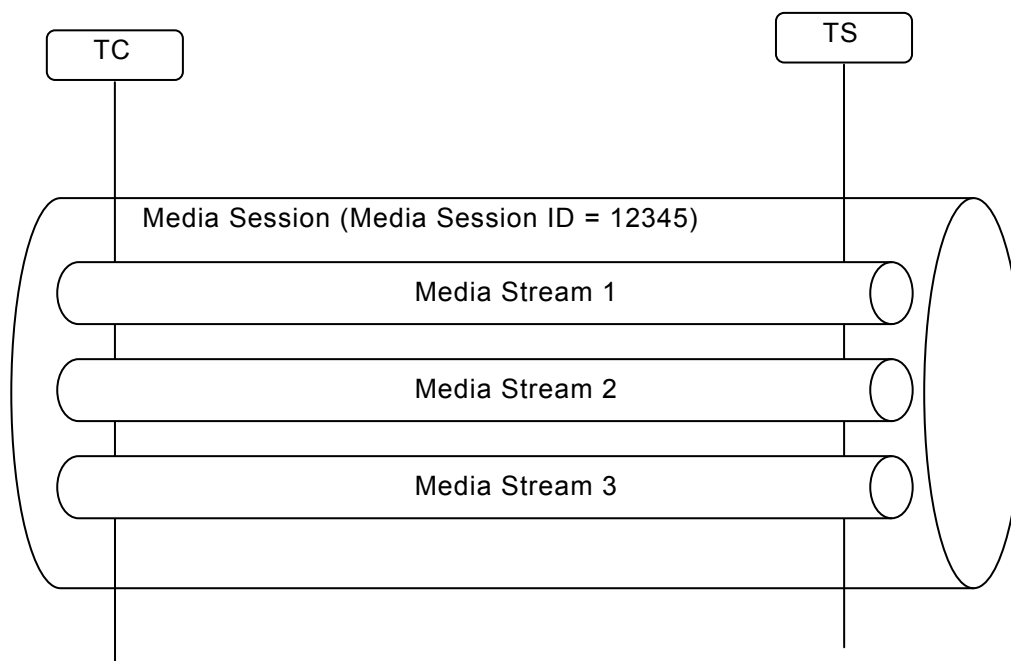


Figure 1 — Relationship among Media Stream, Media Session and Media Session ID

3.3.7**Media Capability**

Media Stream's attributes which are negotiated before sending/receiving a Media Stream in a Media Session. Media Stream's attributes include a codec type, a codec parameter, IP address, and a port number etc. [7] defines one of the methods to describe Media Capability. And the Media Capabilities represent attributes for a set of Media Streams.

3.3.8**Caller**

terminal which initiates a Call.

3.3.9**Callee**

terminal which receives a Call.

3.3.10

Remote Party

Callee, relative to the Caller

Caller, relative to the Callee

3.3.11

Peer

participant in a Call.

3.3.12

Monopolize

to obtain the exclusive rights to control a specific Call managed by the [CallManagement](#) service. When a Call is successfully Monopolized by an identity, the [CallManagement](#) service only allows this specific identity to manage the Call, until the Call is terminated or this identity hands over the exclusive rights to control the Call to another identity.

3.3.13

Monopolizer

identity (a TelCP or a user) that currently has the exclusive rights to control a specific Call managed by the [CallManagement](#) service.

3.3.14

Call Monopolization

mechanisms that allow an identity (a TelCP or a user) to Monopolize a Call managed by the [CallManagement](#) service. See 5.2.2 in detail.

3.3.15

Monopolization Mode Call

Call which is Monopolized.

3.3.16

Non-Monopolization Mode Call

Call which is not Monopolized. Therefore, any TelCP can control a Non-Monopolization Mode Call.

3.3.17

TelCPName

unique name for the TelCP.

3.3.18

Secret Key

unique value assigned to a TelCP. The [CallManagement](#) service uses the Secret Key to verify whether an authorized TelCP is invoking the action.

3.3.19

Push Information

feature that enables a TS to event information received from a service provider on WAN side. This feature also allows to event some status changes that occurs within a TS.

3.3.20

Early Media

media (e.g., audio and video) that is sent and/or received before a Call is accepted or rejected by a Callee. Typical examples of Early Media generated by a Callee are ringing tone and announcements (e.g., queuing status). Early Media generated by a Caller typically consists of voice commands or dual tone multi-frequency (DTMF) tones to drive interactive voice response (IVR) systems.

3.3.21**Content Sharing**

feature that enables a TS or TC to share content (e.g., audio, video or image etc.) to the Remote Party using the Media Session during a Call. See 5.2.6 for more detail.

3.3.22**Parallel Call**

secondary Call established between the devices which are selected by the Peers already participating in a regular telephony Call.

3.3.23**Parallel Call Setup**

process to exchange identities of terminals which will establish Parallel Call.

3.4 Abbreviated terms**3.4.1****GUI**

Graphical User Interface

3.4.2**ID**

Identifier

3.4.3**SDP**

Session Description Protocol

3.4.4**TC**

Telephony Client

3.4.5**TelCP**

Telephony Control Point

3.4.6**TS**

Telephony Server

4 Notations and conventions**4.1 Text conventions**

- Strings that are to be taken literally are enclosed in “double quotes”.
- Words that are emphasized are printed in *italic*.
- Keywords that are defined by the UPnP Working Committee are printed using the *forum* character style.
- Keywords that are defined in [1] are printed using the *arch* character style.

4.2 Data Types

This specification uses data type definitions from two different sources. Data types from [1] are used to define state variable and action argument data types. The XML Schema data types [9] are used to define property data types.

For UPnP Device Architecture defined Boolean data types, it is strongly recommended to use the value “0” for false, and the value “1” for true. The values “true”, “yes”, “false”, or “no” may also be used but are not recommended. The values “yes” and “no” are deprecated and shall not be sent out by devices but shall be accepted on input.

For XML Schema defined Boolean data types, it is strongly recommended to use the value “0” for false, and the value “1” for true. The values “true”, “yes”, “false”, or “no” may also be used but are not recommended. The values “yes” and “no” are deprecated and shall not be sent out by devices but shall be accepted on input.

4.3 Vendor-defined Extensions

Whenever vendors create additional vendor-defined state variables, actions or properties, their assigned names and XML representation shall follow the naming conventions and XML rules as specified in [1], 2.5, “Description: Non-standard vendor extensions”.

5 Service Modeling Definitions

5.1 Service Type

The following service type identifies a service that is compliant with this specification:

urn:schemas-upnp-org:service:CallManagement:2

CallManagement service is used herein to refer to this service type.

5.2 CallManagement Service Architecture

This service provides the features for a TelCP to access telephony features that a TS exposes in the UPnP network in order to initiate a Call, accept an incoming Call or receive notifications for incoming Calls, and manage Media Session associated with the Call including starting and stopping the Media Session.

The architecture for the CallManagement service is shown in Figure 2.

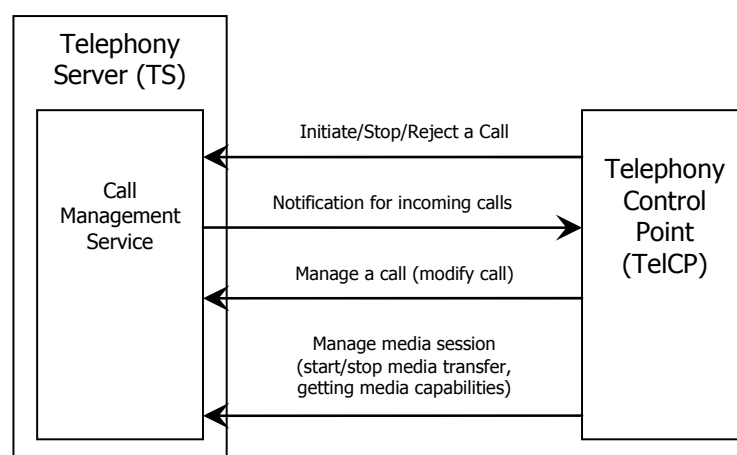


Figure 2 — Architecture of the CallManagement Service

It is assumed that the TelCP features are implemented in a physical device that provides a Graphical User Interface (GUI), for a user to manage locally the telephony functionalities. These are however out of the scope of the CallManagement service design.

The UPnP device that implements the CallManagement service is capable of exposing telephony features to the Peers through some telecommunication means (e.g., PSTN/ISDN, PLMN, and VoIP etc.): The implementation of these telephony features with Peers networks and Peers are out of the scope.

5.2.1 Managing Call With Multiple Telephony Clients

The main components of the telephony architecture are the Telephony Server (TS), Telephony Client (TC), and Telephony Control Point (TelCP). The TS is responsible for interfacing between the Caller or Callee in the wide area network and the TC in the home network. The

multiple Call architecture explained in 5.2.1 considers multiple TCs involve in a telephony Call. When multiple clients in the home network are associated with a Call involving a Callee/Caller in the wide area network, there is a need to share Media Stream among the the clients within the home network so that each client will receive Media Stream from all other clients in addition to the Media Stream from the Callee/Caller in the wide area network. The TS multiplexes Media Streams from multiple TCs in the home network and sends the multiplexed Media Stream to the WAN side.

Figure 3 shows two telephony clients are involved in a Call managed by the TS. The assumption here is that the TCs are in different rooms in the home network and are providing media for the Call with a Caller/Callee in the wide area network which gets media from both TC 1 and TC 2, TC 1 and TC 2 also need to get media from each other in order for the users behind them to listen to each other.

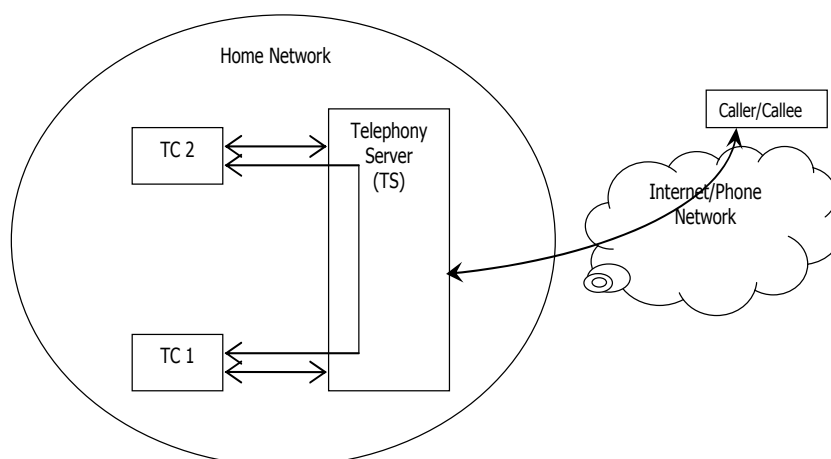


Figure 3 — Managing a Telephony Call with Multiple Telephony Clients

There are two different architectures that can manage Calls with multiple telephony clients. These are:

- a) The TS must make sure that the Media Streams from all the clients including the Callee/Caller in the wide area network are available to each TC. The TS can realize such a feature by performing the mixer functionality where one or more input Media Streams are combined to produce one or more output streams directed specifically to one or more TC. In another words, the TS multiplexes multiple Media Streams from different TCs involved in the Call and also media from the WAN side and sends the multiplexed Media Streams to the other TC involved in the Call. The architecture for sharing Media Stream among multiple telephony clients follows the star architecture where all the TCs are connected to the centralized TS in a star like fashion. The mixer is a software/hardware component resides inside the TS and is not visible in the UPnP network. Its functionality is to gather media inputs from multiple clients and produce output streams directed to each TC. In this approach, while initiating or accepting a Call, the TelCP can decide whether a TC will receive Media Stream from other TCs and whether it is capable of decoding the media from other clients by examining the Media Capability of the clients involved in the Call. Once the TelCP determines that the TC can receive Media Stream from other TCs, the TelCP indicates that option to the TS within the parameter of the StartMediaTransfer (CallID, MediaSessionID, MediaCapabilityInfo) action. The A_ARG_TYPE_MediaCapabilityInfo state variable contains a XML tag that indicates whether the TS should mix media from other clients. The tag is: `<enableMediaMixing> 1 </enableMediaMixing>`. The value of this XML element is either 1 or 0. Where 1 indicates that that TS must mix Media Streams from other clients and a value of "0" indicates that the TS must stop mixing media from other TCs. Figure 4 and Figure 5 show how to use media mixing feature in starting a Call and modifying a Call. Figure 5 shows a scenario where the initial Call was established with only one client TC1. The Call is then modified by adding another client TC2 by invoking the ModifyCall() action and soon after the media

mixing capability is enabled on the TS by invoking the *StartMediaTransfer()* action. Once media mixing is enabled, the TS starts sending multiplexed media from to the TCs. The media mixing capability can be disabled or enabled at any time on the TS by invoking the *ModifyCall()* action with the value of the <enableMediaMixing> element set to either “0” or “1” respectively in the *A_ARG_TYPE MediaCapabilityInfo* input argument.

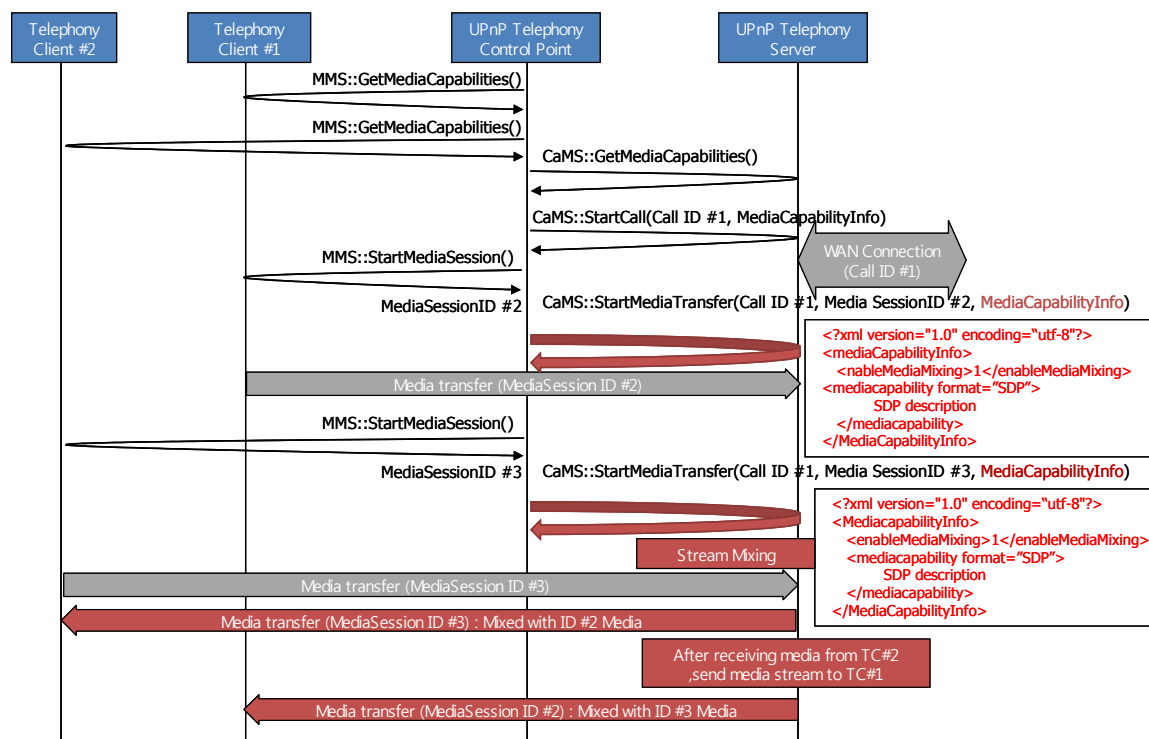


Figure 4 — Starting a Call with Multiple Telephony Clients Using Media Mixing Capability

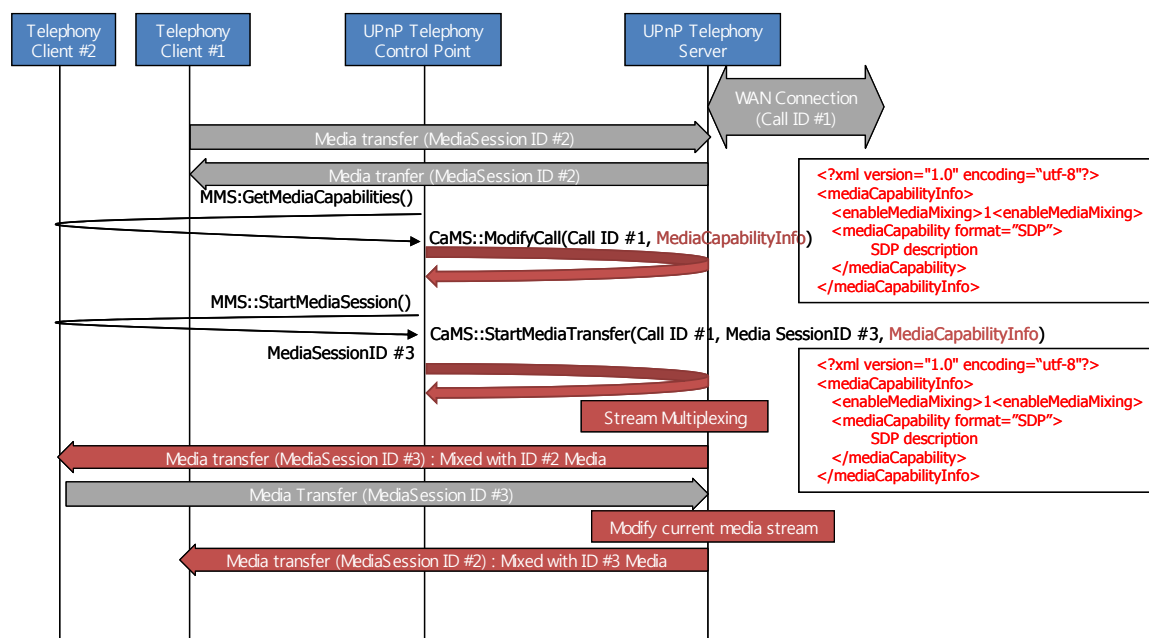


Figure 5 — Modifying a Call with Multiple Telephony Clients Using Media Mixing Capability

- b) An alternative architecture is to create multiple Media Sessions to share Media Streams among the TCs. The TelCP controls the multiple TCs and the TS in order to create

multiple sessions with each TCs. The second architecture requires a TelCP to create multiple Media Sessions. If there are two clients as shown in Figure 6 then the TelCP needs to create 4 different Media Sessions (Session 1 is between TC 1 and TS, Session 2 is between TC 2 and TS, Session 3 is between TS and TC 1 to receive media from TC 2, Session 4 is between TS and TC 2 to receive media from TC 1). The TelCP indicates to the TS whether a session is to share media from another TC by use of a XML tag in the A ARG TYPE MediaCapabilityInfo for the action StartMediaTransfer (CallID, MediaSessionID, MediaCapabilityInfo). The tag is: <shareMedia> Media Session ID </shareMedia>. The Media Session ID identifies the Media Stream to share in a Media Session with a TC. For example, if there are 2 telephony Clients TC1 and TC2 and the Media Session ID for TC1 and TS is 2, and Media Session ID for TC2 and TS is 3, then to create a new Media Session with Media Session ID 4 between TC 1 and TS to get media from TC2, the element will have the following value: <shareMedia>3 </shareMedia>. Similarly, to create a new Media Session with Media Session ID 5 between TC2 and TS and to get media from TC1, the element value will be: <shareMedia> 2 </shareMedia>. Figure 6 and Figure 7 show how to use media sharing feature while starting a Call or modifying a Call.

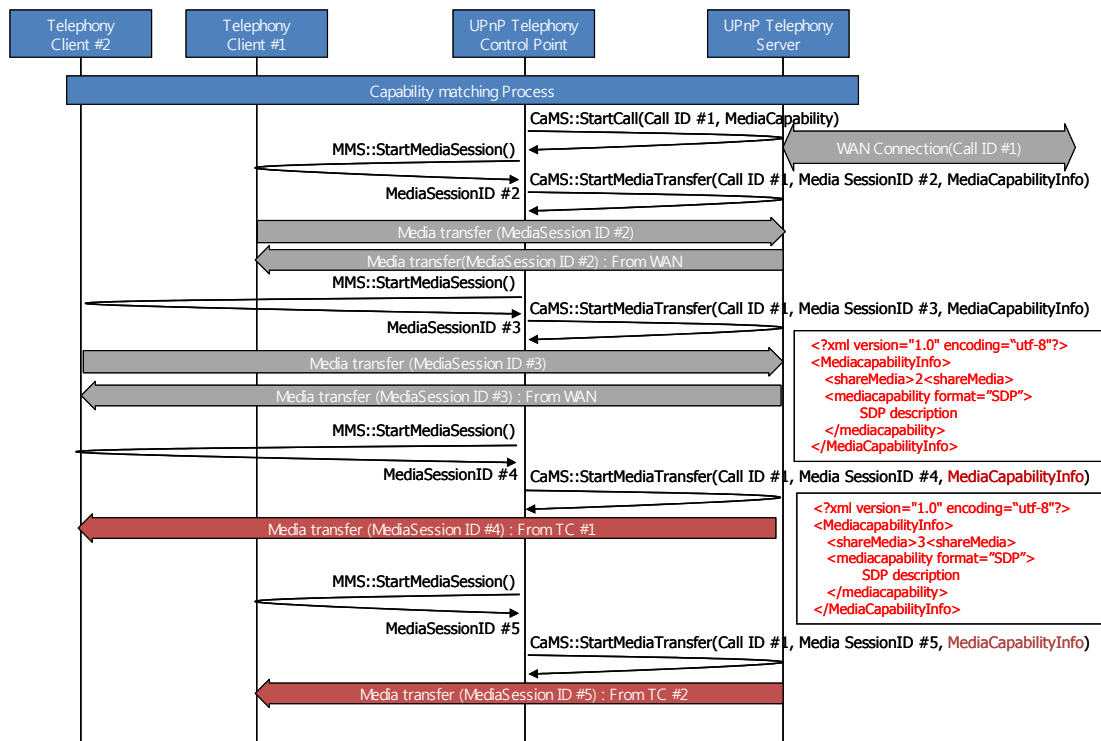


Figure 6 — Starting a Call With Multiple Telephony Clients Using Media Sharing Capability

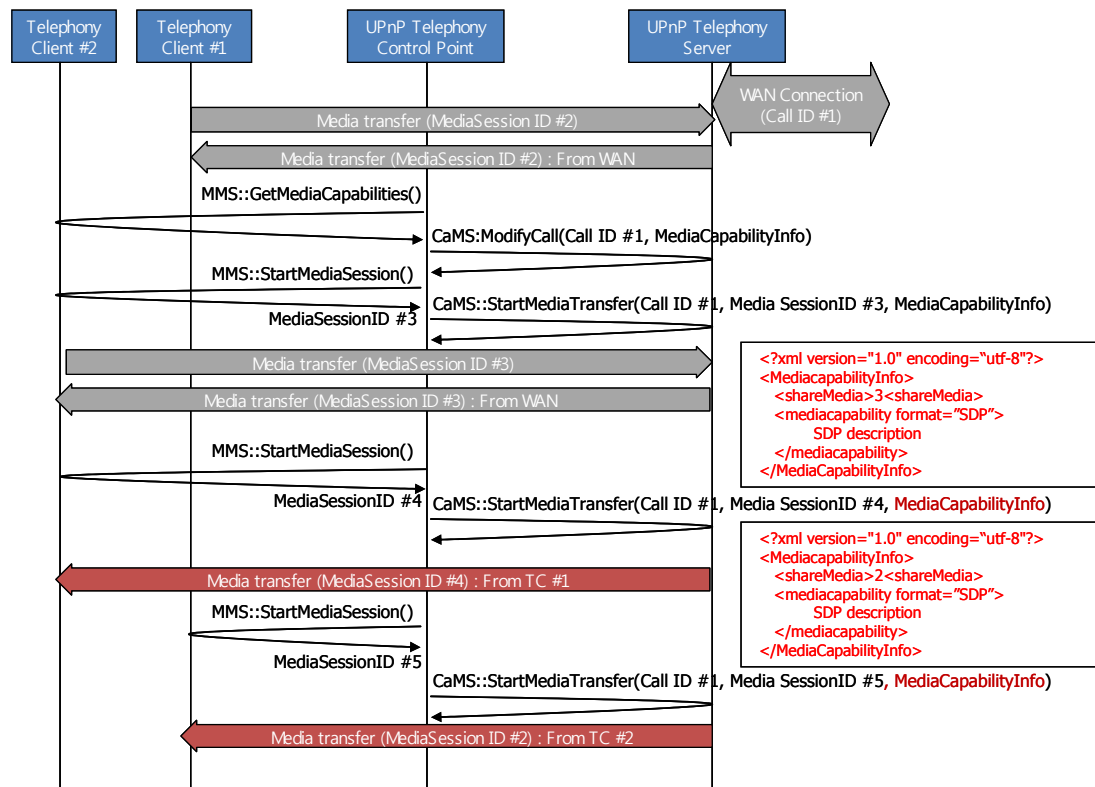


Figure 7 — Modifying a Call With Multiple Telephony Clients Using Media Sharing Capability

5.2.2 User Concept and Multi-User Control

The call management specification provides a mechanism for a Telephony Control Point (TelCP) to register itself with a Telephony Server (TS). The [RegisterTelCPName\(\)](#) action serves this purpose. The action takes the name of the TelCP as an input argument and in response returns a secret key (id for the TelCP). The user concept in this specification can be realized with the same action as used to register a user with the TS. In this instance, the TelCP name will represent the name of the user and the returned secret key which will be the id/secret key for the user. This action will allow a control point to set a unique name for the user. A control point can register multiple users and in each case the TelCP name will represent a different user which will have a different name.

When a TS receives an incoming Call or an SMS or a presence request, the server sends a notification message. The notification message may not include details of the Call and depends on the policy set in the server. The notification message may also contain the name of the user for whom the Call is directed. The notification message can also include the names of TelCP. The “type” attribute of the [targetNames](#) element of the [CallInfo](#) state variable can determine whether the message is directed for a user or for a TelCP.

The TelCP monitors the events from the TS. If a TelCP recognizes a user identifier in the notification message from a previous registration the control point can respond with an action to obtain Call information from the TS. The TelCP where the user is located, invokes the action by providing the key/id of the user. These actions may be optionally sent over a UPnP secure session with the TS to protect inbound Call information or presence information transmitted on the home-network from being intercepted or modified.

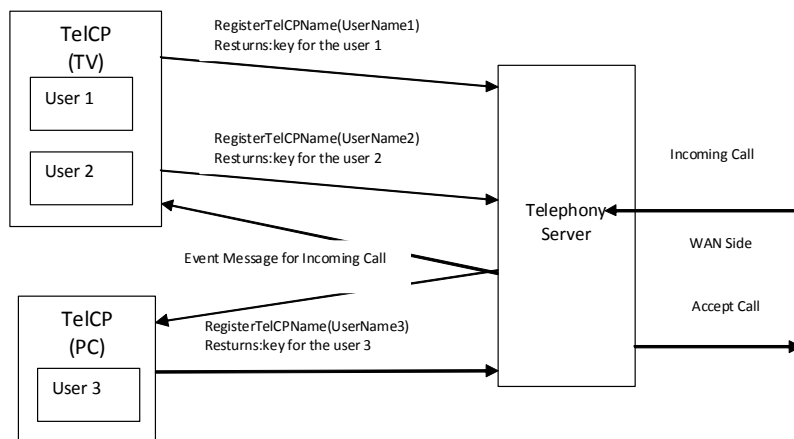


Figure 8 — Example for Multi-User Control Concept

Figure 8 shows a TV with a TelCP. The TelCP is managing two users. The following process explains how the multi-user call management system works:

- The TelCP on the TV registers both users with the Telephony Server. The Telephony Server returns the key for the user for each registration action.
- Each user can log onto the TV using their user name and password. However, the login mechanism is application specific and out of scope of this specification.
- The Telephony Server can be configured locally for different users. The configuration includes setting up which calls will be controlled by which user and by which control point etc.
- When a TelCP receives an event for notification of an incoming Call from the Telephony Server, the TelCP check the notification message. The body of the notification message may contain the name of the users for whom the Call is intended.
- If the Call is for one of the users which the TelCP is managing, the TelCP invokes an action on telephony server to accept or reject the Call.
- When invoking the action to accept/reject, the TelCP also supplies the key associated with the user for whom the Call is intended.
- The Telephony Server will only accept the request from a TelCP that has supplied the valid key and is in the list of authorized TelCP/User to invoke this action.
- The above process is not only valid for managing Call but also for managing presence, voice mail or other telephony services.

5.2.3 Call Monopolization

The telephony architecture enables a Telephony Control Point (TelCP) to manage Calls on the Telephony Server (TS). This includes instructing the TS to start an outgoing Call or answer an incoming Call with a Callee in the wide area network, or instructing the TS to modify or terminate an existing Call with the Callee. In practical deployment scenarios, there can be more than one TelCP in the home network (e.g., multiple TVs and PCs in difference rooms that have TelCP functions). The basic UPnP architecture allows any control point to control UPnP devices. However, in some situations, a user may require important Calls (e.g. private Calls) to be managed only by a limited number of TelCPs. For example:

- a) When a Call on the TS is initiated or accepted by a TelCP which is controlled by a specific user, the user may require this Call to be managed (i.e. modified or terminated) afterwards only by this specific TelCP, or by the user itself by using other TelCPs.
- b) When the TS receives an incoming Call, the user may require that the Call can only be answered by a specific TelCP or a user (i.e., any attempt to answer this Call by unauthorized TelCP, or by an unauthorized users using a TelCP will fail).

Subclause 5.2.2 addresses the requirement of a). The term Call Monopolization refers to a mechanism that allows an identity (a TelCP or a user) to manage a Call. Depending on the type of identity (i.e. TelCP or user), the Call Monopolization can be performed at the “TelCP-level” allowing the Call to be managed by a specific TelCP; or at the “User-level” allowing the Call to be managed only by a specific user using any TelCP.

The CallManagement service is capable of handling the incoming Calls meeting the requirement of b). When receiving an incoming Call, the CallManagement service determines by its internal policies whether any TelCP or user can answer the Call. If not, the service announces via the evented CallInfo state variable which one or more IDs (TelCPNames or usernames) are allowed to answer (accept or reject) the Call. Depending on the type of identity (i.e. a TelCP or a user), the incoming Call can be handled at the “TelCP-level” allowing the incoming Call to be answered by a specific TelCP(s); or at the “User-level” allowing to be answered by a specific user(s) using any TelCP.

There are two mechanisms for Call Monopolization defined in this version of the specification. Subclause 5.2.3.1 defines the mechanism called “PHONE-based Call Monopolization”, targeting only TelCP-level Call Monopolization. When a TS implements the DeviceProtection service, this version of the specification defines another mechanism for Call Monopolization which utilizes the DeviceProtection service and are defined in [15] and named as “DP-based Call Monopolization”. This mechanism enables both TelCP-level and User-level Call Monopolization.

5.2.3.1 PHONE-based Call Monopolization

The PHONE-based Call Monopolization defines a mechanism for a specific TelCP to Monopolize a Call managed by the CallManagement service. The CallManagement service defines the A_ARG_TYPE_TelCPName state variable, which is used by this service to identify a TelCP. This service also defines the A_ARG_TYPE_SecretKey state variable, which is used by this service to verify the TelCP’s identity. When a new TelCP starts up, it is required to Call the RegisterTelCPName() action defined in this service to register a unique TelCPName for itself. The action returns a Secret Key which is used subsequently by this service to verify the TelCP’s identity when the TelCP tries to Monopolize a Call or manage a Monopolization Mode Call. Figure 9 and text below illustrates how the PHONE-based Call Monopolization mechanism enables a TelCP to Monopolize a Call.

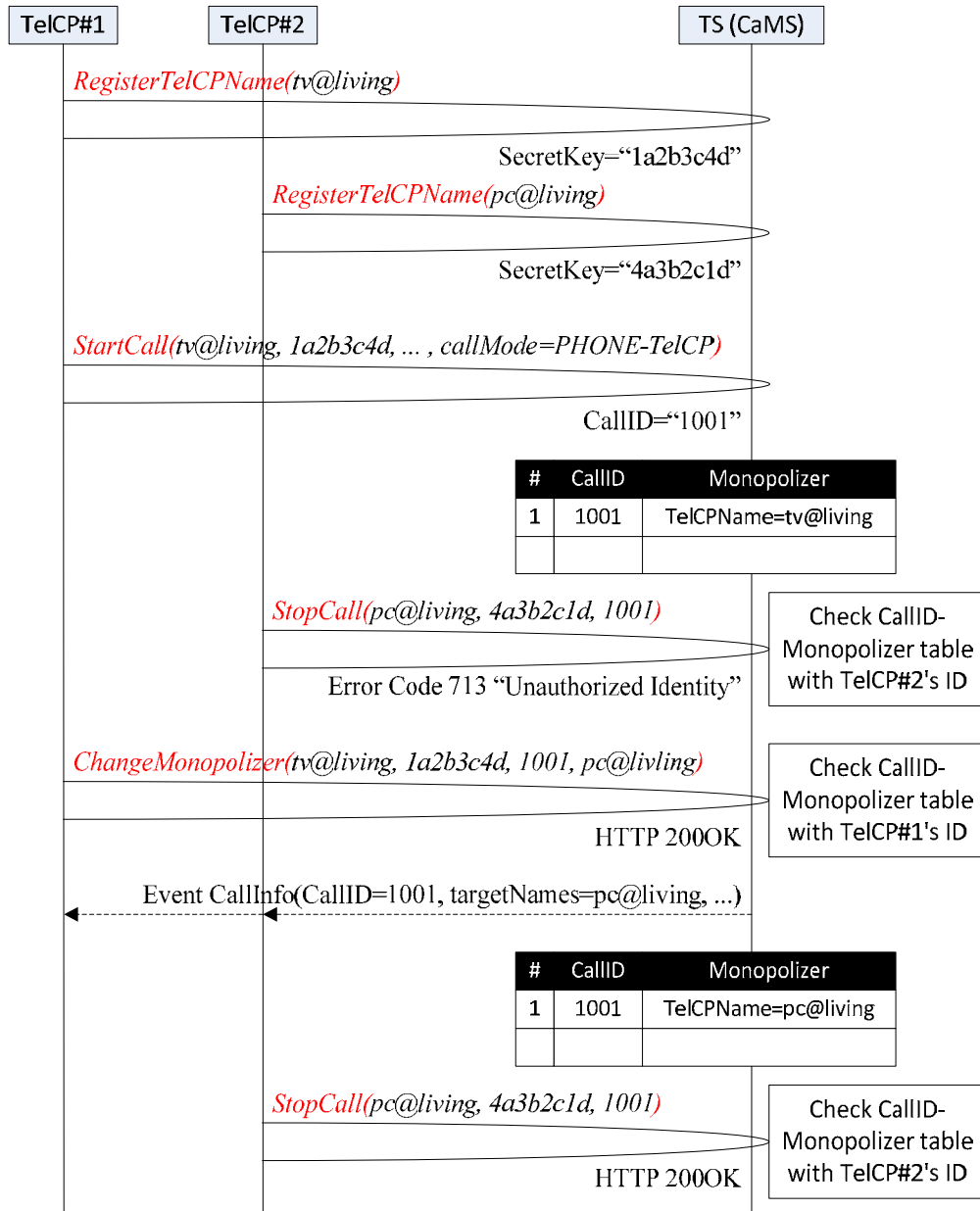


Figure 9 — PHONE-based Call Monopolization

- If a TelCP, e.g., TelCP#1 in Figure 9 wants to Monopolize a Call managed by the CallManagement service, it must first have registered a TelCPName with the service. A new TelCP can invoke the RegisterTelCPName() action to register a unique TelCPName. This action will assign a Secret Key for the TelCP.
- The TelCP#1 can instruct the CallManagement service to initiate a Call by invoking the StartCall() action. In order to Monopolize this specific Call, the TelCP must invoke the action as follows:
 - The input arguments TelCPName and SecretKey of this action must indicate the right values of the TelCP#1's TelCPName and Secret Key, which in this example are "tv@living" and "1a2b3c4d" respectively.
 - The CallMode input argument of this action must also indicate that the Call is to be managed in Monopolized mode by using the "PHONE-TelCP" as its value.
- The CallManagement service verifies the identity of TelCP#1 and if passed, returns the CallID and recognize the TelCP#1 as the current Monopolizer of the Call identified by this

ISO/IEC 29341-26-10:2017(E)

CallID. The CallManagement service internally maps the CallID with Monopolizer's ID (i.e. TelCP#1's TelCPName).

- d) Subsequently, any management action on this Call (e.g., ModifyCall() or StopCall()) must be checked by the CallManagement service to make sure if the action is invoked by the Monopolizer of this Call. For example, in Figure 9,
- TelCP#2 which is not the current Monopolizer of the Call identified by CallID=1001 is not allowed to stop the Call by the CallManagement service (i.e., the StopCall() action will be returned with Error Code 713 "Unauthorized Identity".
- e) The CallManagement service also provides an interface for a Call Monopolizer (i.e. TelCP) to hand over the management rights of the Call to another TelCP.
- By invoking the ChangeMonopolizer() action, the TelCP#1 as the Monopolizer of the Call identified by CallID=1001 hands over the management rights to TelCP#2.
 - The CallManagement service checks TelCP#1's identity and also the CallID by checking internal Monopolizer mapping table, if successful, the table will be updated to indicate TelCP#2 as the current Monopolizer of this Call. The service also events the updated CallInfo state variable indicating the new Monopolizer of this Call.
 - TelCP#2, as the new Monopolizer, is allowed to manage this Call by the CallManagement service, while TelCP#1 is not allowed to manage the Call anymore.

5.2.3.1.1 Incoming Call Handling

When the CallManagement service receives an incoming Call, it can decide by its internal policies whether only a subset of TelCP(s) is allowed to answer the Call. Since the registered TelCPName is used to identify a TelCP, the service announces the allowed identities to answer the incoming Call by listing the relevant TelCPNames as the value of <targetNames> element in the evented CallInfo state variable. Note that the policies itself and the way to configure them are out of the scope of this version of the CallManagement service. When a TelCP invokes the AcceptCall() or RejectCall() action targeting on the Call, the CallManagement service will verify the TelCPName and Secret Key of the TelCP to determine if it is allowed to answer the Call. Figure 10 and text below illustrates how the CallManagement service handles an incoming Call.

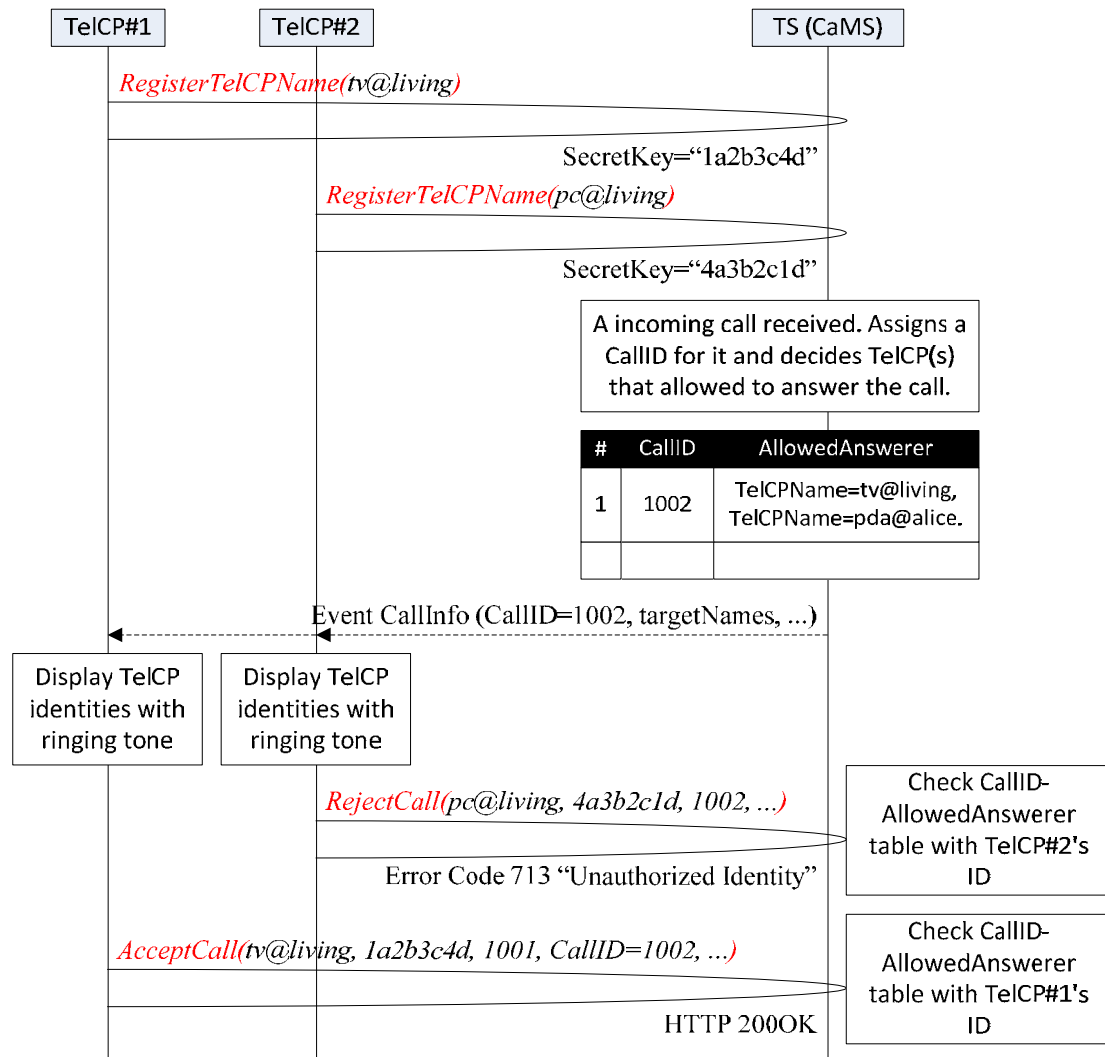


Figure 10 — Incoming Call Handling

- If a TelCP, e.g., TelCP#1 in Figure 10 wants to become a candidate to answer an incoming Call managed by the *CallManagement* service, it must first have a registered TelCPName in the service. A new TelCP can invoke the *RegisterTelCPName()* action to register a unique TelCPName. The action will assign a Secret Key for the TelCP.
- When the *CallManagement* service receives an incoming Call, it can decide whether one or more TelCPs are allowed to answer the Call. The service also assigns a unique CallID for this Call. The service internally maps the CallID with the TelCPName that can manage the Call.
- The *CallManagement* service sends the *CallInfo* state variables of the incoming Call to the TelCPs that had subscribed to the events for this service. The evented *CallInfo* state variable contains the CallID of the Call as well as the TelCPNames of the TelCPs that are allowed to answer the Call.
- When receiving the *AcceptCall()* or *RejectCall()* action from a TelCP the *CallManagement* service verifies the identity of the TelCP by checking its TelCPName and Secret Key from the input arguments of the action. The service also looks up in the "CallID-AllowedAnswerer" table with the TelCPName to decide if the TelCP is allowed to answer/reject the Call. For example, in Figure 10,
 - TelCP#2 with "pc@living" as its TelCPName is not one of the allowed TelCP that can answer the incoming Call, thus the action invoked by this TelCP targeting the Call will fail.

- TelCP#1 has its TelCPName “tv@living” in the evented [CallInfo](#) state variable, thus it will succeed in answering the Call by invoking the [AcceptCall\(\)](#) action.

5.2.4 Selective Information Delivery

The [CallManagement](#) service defined in this specification provides features enabling privacy. For example, when a user starts calling with a Callee in the wide area network, he/she may require some information (e.g. the peer ID) of this Call to be unavailable to the other users in the home network. To meet the above requirement, “Selective Information Delivery” mechanism is used by the [CallManagement](#) service.

The Selective Information Delivery defines a mechanism by which the [CallManagement](#) service only delivers general information (e.g. Call ID) of a specific Call to the TelCPs, so that privacy related information (e.g. peer ID) are hidden from unauthorized TelCPs or users.

When a Call is managed by the service is Monopolized, or when the service decides that only specific TelCP(s) or User(s) can answer an incoming Call, the Selective Information Delivery mechanism is applied.

- The [CallManagement](#) service events the [CallInfo](#) state variable to all the subscribed TelCPs containing only the general information of the Call, such as [<callID>](#), [<targetNames>](#), and [<callStatus>](#) elements. See 5.3.2 for more detail on which elements of the [CallInfo](#) state variable should be and should not be evented.
- When the [CallManagement](#) service receives the [GetCallInfo\(\)](#) action request targeting the Call, the [CallManagement](#) service verifies if the current identity (TelCP or User) is the Monopolizer or allowed to answer the Call. If so, the returned [CallInfo](#) state variable contains full information of the Call; otherwise, it contains only general information of the Call, such as [<callID>](#), [<TelCPNames>](#), and [<callStatus>](#) elements. See 5.3.2 for more detail on which elements of the [CallInfo](#) state variable should be and should not be returned.

5.2.5 Handling of Media Streams

The [CallManagement](#) service provides two kinds of mechanisms for handling the Media Streams between the Peers. The first mechanism is “TC-Based Media Handling” and the other mechanism is “TS-Based Media Handling”.

5.2.5.1 TC-Based Media Handling

In the TC-Based Media Handling the TelCP uses one or more TC devices in the ongoing Call. The TelCP creates Media Session(s) between the TS and the TC(s). The Media Streams are sent and/or received between the TC(s) and the Remote Party via the TS using the Media Session(s). For establishing the Media Session for the Call the TelCP negotiates a common set of Media Capabilities between the TS and the TC by matching the Media Capabilities supported by the TS and the TC(s). The TelCP can get the list of supported Media Capabilities of the TS by invoking the [CaMS::GetMediaCapabilities\(\)](#) action or by receiving the [CallInfo](#) state variable event when the TS receives an incoming Call request or a modify Call request. The Media Capabilities of the TS are described in the [<mediaCapability>](#) element of the [A_ARG_TYPE MediaCapabilityInfo](#) state variable or the [CallInfo](#) state variable which are defined in this document. The TelCP can get the list of supported Media Capabilities of the TC by invoking the [MMS::GetMediaCapabilities\(\)](#) action. The Media Capabilities of the TC are described in the [<mediaCapability>](#) element of the [A_ARG_TYPE MediaCapabilityInfo](#) state variable which is defined in the [MediaManagement](#) service [10].

Once the TelCP decides on the Media Capabilities for the Call, the TelCP invokes the [StartCall\(\)](#), [AcceptCall\(\)](#), [ModifyCall\(\)](#) or [AcceptModifyCall\(\)](#) action to negotiate the Media Capabilities with the Remote Party for establishing the Call or modifying the ongoing Call. When the Call is successfully established or modified, the TelCP invokes the [MMS::StartMediaSession\(\)](#) action on the TC and the [CaMS::StartMediaTransfer\(\)](#) action on the TS to start the Media Stream transfer between the TS and TC(s). The detail sequence diagrams for the TC-Based Media Stream handling are shown in C.1.

5.2.5.2 TS-Based Media Handling

A TS can be implemented on a physical device like a mobile phone or a HGW. More often physical device itself has the capabilities to handle the Media Streams with certain media codec support. This kind of devices also provide input/output functionality for the media transfer (e.g., a mobile phone has an audio codec, capability to send and/or receive an audio stream, and mic/speaker as the input/output devices to talk with the Remote Party during the Call). The TS-Based Media Handling enables a TelCP to use the TS capabilities of handling the Media Stream in a Call without implementing the TC device functionalities on the physical device which has the TS. For this purpose, the TS exposes the Media Capabilities corresponding to the Media Streams which can be handled by the TS. These Media Capabilities are referred to as the “Native Media Capability” of the TS and are described in the `<nativeMediaCapability>` element of the [CallInfo](#) state variable or in the [A_ARG_TYPE_MediaCapabilityInfo](#) state variable. A TelCP can get the Native Media Capabilities of the TS by invoking the [GetMediaCapabilities\(\)](#) action or receiving the [CallInfo](#) state variable when the TS receives an incoming Call or a modify Call request. The TelCP decides on the Native Media Capability for the Call and then the TelCP invokes the [StartCall\(\)](#), [AcceptCall\(\)](#), [ModifyCall\(\)](#) or [AcceptModifyCall\(\)](#) action to negotiate the Media Capabilities with the Remote Party for establishing the Call or modifying the ongoing Call. When the Call is successfully established or modified, the TelCP invokes [CaMS::StartMediaTransfer\(\)](#) action on the TS to start the Media Stream transfer between the TS and the Remote Party. The detail sequence diagrams for the TS-Based Media Stream handling are shown in C.1.

The “TC-Based Media Handling” and “TS-Based Media Handling” can be used simultaneously in a Call. For example, when the TS is implemented on a mobile phone and there is a TC for sending and/or receiving a video stream, the TelCP can create a video Call using the mobile phone as a TS for sending and receiving the audio stream and the TC for sending and/or receiving the video stream. In this case, the TelCP uses “TC-Based Media Handling” for the video capabilities and “TS-Based Media Handling” for the audio capabilities.

5.2.6 Content Sharing

Content Sharing is a feature that enables sharing of content between a TS and the Remote Party during a Call. The users can share audio/video or image content to the Remote Party when talking. The Content Sharing feature utilizes the Media Session between the Peers to transport the content. The content can be sourced from UPnP [MediaServer](#) devices, or locally from TC/TS, or from non-UPnP media serving devices. Based on the commands from a TelCP, the selected content is retrieved, processed by the TC or TS, and then shared to the Remote Party by the TS. There are two types of Content Sharing mechanisms that are described in subclauses 5.2.6.1 and 5.2.6.2.

5.2.6.1 TS-Based Content Sharing

The TS-Based Content Sharing mechanism allows a TS to retrieve, process and then send the content to the Remote Party during a Call. The mechanism requires TS to have more media processing capabilities, including retrieving media from media serving devices and transcoding. The TS supporting this mechanism should be able to:

- Retrieve content from the content hosting device, e.g. using HTTP GET method to retrieve the content from a UPnP [MediaServer](#) device.
- Process the content before sending to the Remote Party, e.g. transcode to the codec used for the Call, mix the audio content with the speech media.
- Update or use the existing Media Session of the Call to send the content, e.g. renegotiate the Media Capability with the Remote Party to add new channels to transport the A/V content, or send the mixed media in the current channel.

A TelCP supporting this feature shall be able to locate the content (e.g., having AV CP functionalities) and provide the TS with the content information, and then trigger the TS to fetch the content and share to the Remote Party using the actions of this service. The TS supporting this feature shall implement the `<contentSharingCapability>` element defined

ISO/IEC 29341-26-10:2017(E)

in 5.3.12. C.5.1 provides examples on detailed sequence for sharing content during a Call using the TS-Based Content Sharing mechanism.

5.2.6.2 TC-Based Content Sharing

The TC-Based Content Sharing mechanism allows a TC to retrieve, process and then send the content to the Remote Party via a TS during a Call. The mechanism requires a TC to participate in a Call and be responsible for fetching and processing the media content, and sending it to the Remote Party via a TS using the Media Session with the TS. The TC supporting this mechanism should be able to:

- Retrieve content from the content hosting device, e.g. using HTTP GET method to retrieve the content from a UPnP [MediaServer](#) device.
- Process the content before sending to the TS, e.g. transcode to the codec used for the Call, mix the audio content with the speech media.
- Update or use the existing Media Session to send the content to the TS, e.g. add new Media Streams to transport the A/V content, or send the mixed media in the current Media Streams.

A TelCP supporting this feature shall be able to locate the content (e.g., having AV CP functionalities) and provide the TC with the content information, instruct the TS and the TC to update media sessions in the WAN and LAN to add new streams if necessary for sending the content. The TC supporting this feature shall implement the <contentSharingCapability> element defined in the [MediaManagement](#) service in [10]. C.5.2 of this document provides examples on detailed sequence for sharing content during a Call using the TC-Based Content Sharing mechanism.

5.3 State Variables

Note: For first-time reader, it may be more insightful to read the theory of operations first and then the action definitions before reading the state variable definitions.

5.3.1 State Variable Overview

Table 1 — State Variables

Variable Name	R/A ^a	Data Type	Reference
CallInfo	A	string (XML fragment)	See 5.3.2
TelCPNameList	A	string	See 5.3.3
CallBackAvailability	A	string (XML fragment)	See 5.3.4
PushInfo	O	string (XML fragment)	See 5.3.5
VoiceMailInfo	O	string (XML fragment)	See 5.3.6
ParallelCallInfo	O	string (XML fragment)	See 5.3.7
A_ARG_TYPE_TelephonyServerIdentity	R	String	See 5.3.8
A_ARG_TYPE_TelCPName	A	String	See 5.3.9
A_ARG_TYPE_TelCPNameList	A	String	See 5.3.10
A_ARG_TYPE_Expires	A	i4	See 5.3.11
A_ARG_TYPE_MediaCapabilityInfo	A	string (XML fragment)	See 5.3.12
A_ARG_TYPE_CalleeID	R	string	See 5.3.13
A_ARG_TYPE_CallPriority	A	string	See 5.3.14

Variable Name	R/A ^a	Data Type	Reference
<u>A_ARG_TYPE_CallMode</u>	<u>A</u>	<u>string</u>	See 5.3.15
<u>A_ARG_TYPE_CallID</u>	<u>R</u>	<u>string</u>	See 5.3.16
<u>A_ARG_TYPE_SecretKey</u>	<u>A</u>	<u>string</u>	See 5.3.17
<u>A_ARG_TYPE_RejectReason</u>	<u>A</u>	<u>string</u>	See 5.3.18
<u>A_ARG_TYPE_TCList</u>	<u>A</u>	<u>string</u> (XML fragment)	See 5.3.19
<u>A_ARG_TYPE_CallInfoList</u>	<u>A</u>	<u>string</u> (XML fragment)	See 5.3.20
<u>A_ARG_TYPE_CallLogs</u>	<u>A</u>	<u>string</u> (XML fragment)	See 5.3.21
<u>A_ARG_TYPE_CallBackID</u>	<u>A</u>	<u>string</u>	See 5.3.22
<u>A_ARG_TYPE_CallBackInfoList</u>	<u>A</u>	<u>string</u> (XML fragment)	See 5.3.23
<u>A_ARG_TYPE_PushInfoList</u>	<u>A</u>	<u>string</u> (XML fragment)	See 5.3.24
<u>A_ARG_TYPE_VoiceMailInfoList</u>	<u>A</u>	<u>string</u> (XML fragment)	See 5.3.25
<u>A_ARG_TYPE_VoiceMailID</u>	<u>A</u>	<u>string</u>	See 5.3.26
<u>A_ARG_TYPE_CallType</u>	<u>A</u>	<u>string</u> (XML fragment)	See 5.3.27
<u>A_ARG_TYPE_CallerID</u>	<u>A</u>	<u>string</u>	See 5.3.28
<u>A_ARG_TYPE_MaxWaitingTime</u>	<u>A</u>	<u>l4</u>	See 5.3.29
^a <u>R</u> = required, <u>A</u> = allowed, <u>CR</u> = conditionally required, <u>CO</u> = conditionally allowed, <u>X</u> = Non-standard, add <u>-D</u> when deprecated (e.g., <u>R-D</u> , <u>A-D</u>).			

5.3.2 CallInfo

The format of the CallInfo state variable is an XML document. It includes information about a Call. It contains the CallID, the priority of the Call, Remote Party's ID, Call Status, TC's information (i.e., UUID and Media Session ID), and Media Capability etc.

The CallInfo state variable is evented when the CallManagement service receives an incoming Call or the Call Status of an existing Call is changed.

5.3.2.1 XML Schema Definition

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for CallInfo in the XML namespace "urn:schemas-upnp-org:phone:cams" which is located at "http://www.upnp.org/schemas/phone/cams-v2.xsd".

5.3.2.2 Description of fields in the CallInfo structure

```
<?xml version="1.0" encoding="utf-8"?>
<cams:callInfo
  xsi:schemaLocation="urn:schemas-upnp-org:phone:cams
    http://www.upnp.org/schemas/phone/cams-v2.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:cams="urn:schemas-upnp-org:phone:cams"
  xmlns:peer="urn:schemas-upnp-org:phone:peer">
  <callID>unique ID for the Call</callID>
  <targetNames type="type of the targetName">
    unique names or IDs for targets
  </targetNames>
  <callStatus reason="reason why Call Status is changed">
    status of the Call
  </callStatus>
```

```

<priority>priority of the Call</priority>
<remoteParty>
  <peer:id>ID of the Peer</peer:id>
</remoteParty>
<TCList>
  <TC>
    <UDN>uuid:UUID</UDN>
    <mediaSessionID>Unique Media Session ID</mediaSessionID>
  </TC>
  <!-- Any other TC (if any) go here.-->
</TCList>
<enableMediaMixing>1 or 0</enableMediaMixing>
<shareMedia>Media Session ID</shareMedia>
<mediaCapability format="format of the Media Capability">
  Media Capability
</mediaCapability>
<nativeMediaCapability format="format of the Native Media Capability">
  Native Capability
</nativeMediaCapability>
</cams:callInfo>

```

<xml>

Required. Case Sensitive.

<CallInfo>

Required. Shall include the name space declaration for the complex type <peerType> ("urn:schemas-upnp-org:phone:peer") and the namespace declaration for the [CallManagement](#) service Schema ("urn:schemas-upnp-org:phone:cams"). This namespace "urn:schemas-upnp-org:phone:cams" defines the following elements and attributes:

<callID>

Required. xsd:string, shall include the CallID which is unique for a Call.

The [CallManagement](#) service generates the CallID when this service receives an incoming Call or receives [StartCall\(\)](#) or [InitiateCall\(\)](#) action.

<targetNames>

Required. xsd:string, shall include a list of identities (TelCP or user) for Selective Information Delivery (See 5.2.4). The type of identifiers is decided by the attribute "type". The identifiers are listed by comma separated values. When the value of this element is set to "*", all the identifiers are allowed to manage the Call. An identifier shall not include any comma(",") in the string. The allowed values of this attribute are as follows:

Table 2 — Allowed values for type

Value	Description
" TelCPName "	This identity is described by TelCPName which is defined in the CallManagement service.
" TelCPID "	This identity is described by Control Point Identity which is defined in the DeviceProtection service [16].
" Username "	This identity is described by Username which is defined in the DeviceProtection service [16].

<callStatus>

Required. xsd:string, Indicates the status of the Call. The allowed values of this elements are as follows: And this element has an attribute which name is "reason".

Table 3 — Allowed values for callStatus

Value	Description
" Ringin "	The status of the Call when a TS receives an incoming Call. The status of the Call remains as "Ringin" state until a TelCP accepts or rejects the Call.
" Dialin "	The status of the Call when the TS is initiating the Call. The status of the Call remains as "Dialin" state until the Call is received by the Callee in WAN side.

Value	Description
<u>"Calling"</u>	The status of the Call when the Callee receives the Call. The status of the Call remains as "Calling" state until the Call is answered by the Callee.
<u>"Connected"</u>	The status of the Call when the Call is accepted by the Callee. The status of the Call remains as "Connected" state until the Media Streams transfer start.
<u>"Talking"</u>	The status of the Call when the Media Streams transfer starts. The status of the Call remains as "Talking" state until the Call is terminated or modified.
<u>"Disconnected"</u>	The status of the Call when the Call is terminated.
<u>"SendingModifyRequest"</u>	The status of the Call when the TS sends the request to modify the Media Capability of an existing Call (e.g. adding video capability to the audio Call). The status of the Call remains as "SendingModifyRequest" state until the Remote Party answers the modify request.
<u>"ReceivingModifyRequest"</u>	The status of the Call when the TS receives the request to modify the Media Capability of an existing Call from the WAN side (e.g. adding video capability to the audio Call). The status of the Call remains as "ReceivingModifyRequest" state until the TelCP answers the modify request.
<u>"Modified"</u>	The status of the Call when the modify request is accepted by the Caller or the Callee. The status of the Call remains as the "Modified" state until the Media Streams transfer resumes based on the new Media Capabilities.
<u>"ModifyFailed"</u>	The status of the Call when the modify request is failed or rejected. The status of the Call remains as the "ModifyFailed" state until the Media Streams transfer resumes.

reason

Allowed. xsd:string, Indicates the reason why status of a Call is changed. The following table shows the values of reason attribute and relevant Call Status:

Table 4 — Allowed values for reason and the corresponding values of the Call Status

Value	Call Status	Description
<u>"Accepted by TS"</u>	<u>"Talking"</u>	The Callee accepted a Call on the TS.
<u>"Initiated by TS"</u>	<u>"Talking"</u>	The TS itself created an outgoing Call.
<u>"Busy"</u>	<u>"Disconnected"</u>	The Callee is busy.
<u>"Not Acceptable Capability"</u>	<u>"Disconnected"</u>	The Callee does not support the Media Capability.
<u>"Not Found"</u>	<u>"Disconnected"</u>	The Callee does not exist.
<u>"Rejected by TS"</u>	<u>"Disconnected"</u>	The Callee rejects an incoming Call on TS
<u>"Rejected"</u>	<u>"Disconnected"</u>	The Callee rejects an incoming Call.
<u>"Rejected"</u>	<u>"ModifyFailed"</u>	The Caller or the Callee rejects an incoming modify request.
<u>"Cancelled"</u>	<u>"Disconnected"</u>	The Caller cancelled the Call.
<u>"Cancelled"</u>	<u>"ModifyFailed"</u>	The Caller or the Callee cancelled the modify request.
<u>"Receiving Early Media Response"</u>	<u>"Calling"</u>	The TS has received a provisional response for the Early Media session.
<u>"Early Media Started"</u>	<u>"Calling"</u>	The Early Media session is established.
<i>TBD</i>	<i>TBD</i>	<i>(Specified by UPnP vendors.)</i>

<priority>

Allowed. xsd:string, Indicates the priority of the Call. In the case of a Monopolization Mode Call, this element is not included in the CallInfo state variable. The <priority> element can have the following values.

Table 5 — Allowed values for priority

Value	Description
<i>“Emergency”</i>	Emergency Call.
<i>“Normal”</i>	Regular Call.
<i>TBD</i>	<i>(Specified by UPnP vendors.)</i>

<remoteParty>

Allowed. peer:peerType, Includes the unique ID of the Remote Party. If the Call is anonymous, the value of the <peer:id> elements set to as the string “anonymous”. In the case of a Monopolization Mode Call, this element is not included in the [CallInfo](#) state variable.

<TCList>

Allowed. Includes information about one or more TCs. In the case of a Monopolization Mode Call, this element and subelements are not included in the [CallInfo](#) state variable. It has following sub elements.

<TC>

Allowed, Includes information about a TC. It has two sub elements: <UDN> and <mediaSessionID>.

<UDN>

Allowed, Indicates the unique device name of a TC.

<mediaSessionID>

Allowed. Indicates the unique ID for a Media Session. See [10] in detail.

<enableMediaMixing>

Allowed. Indicates whether a TS mixes Media Streams from multiple TCs involved in the Call. A value of “1” means that the TS mixes Media Streams from multiple TCs involved in the Call. A value of “0” means that the TS does not mix Media Streams from multiple TCs involved in the Call. In the case of a Non Monopolization Mode Call, if the <enableMediaMixing> element is not included in the [CallInfo](#) state variable, then the behavior of this service is the same as the value of the <enableMediaMixing> element is “0”. In the case of a Monopolization Mode Call, the <enableMediaMixing> element is not included in the [CallInfo](#) state variable. But the output argument of the [GetCallInfo\(\)](#) action can include this element if the Monopolizer invokes the [GetCallInfo\(\)](#) action.

<shareMedia>

Allowed. Includes the Media Session ID to identify the Media Session to be shared among TCs involved in the Call. In the case of a Non-Monopolization Mode Call, if the <shareMedia> element is not included in the [CallInfo](#) state variable, it means that a Media Session is not shared among TCs. In the case of a Monopolization Mode Call, the <shareMedia> element is not included in the [CallInfo](#) state variable when the TS events the [CallInfo](#) state variable. But if the Monopolizer invokes the [GetCallInfo\(\)](#) action, this element is included. The <enableMediaMixing> element and the <shareMedia> element are mutually exclusive and if one element exists in the [CallInfo](#) state variable then the other shall not exist.

<mediaCapability>

Allowed. This element is used to represent the Media Capabilities of an incoming Call or Media Capabilities for the Call which are handled using the TC-Based Media Handling mechanism.

format

Allowed. xsd:string, Indicates the format of Media Capabilities. Allowed values of this attribute are as follows:

Table 6 — Allowed values for format

Value	Description
SDP	Media Capability as defined by [7].
<i>TBD</i>	<i>(Specified by UPnP vendors.)</i>

<nativeMediaCapability>

Allowed. This element is used to represent the TS's Native Media Capabilities or Media Capabilities of the Call which are handled using the TS-Based Media Handling mechanism.

format

Allowed. xsd:string, Indicates the format of Media Capabilities for this element. Allowed values of this attribute are shown in Table 6.

5.3.3 TelCPNameList

The TelCPNameList state variable includes the names of all the existing TelCPs. The format of the TelCPNameList state variable is a comma separated values. It is the same as the format of the A_ARG_TYPE_TelCPNameList. When a new TelCP is registered or a TelCP is unregistered or the name of a TelCP is changed, then the TelCPNameList state variable is evented.

5.3.4 CallBackAvailability

This state variable contains the A_ARG_TYPE_CallBackID state variable. This state variable is evented when a Call that is registered for a Call back becomes available. When a Call is registered for a Call back, the TS returns the CallBackID ID. Once the registered Call becomes available from the WAN side, the TS notifies the availability of the Call using this state variable.

5.3.4.1 XML Schema Definition

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for CallBackAvailability in the XML namespace "urn:schemas-upnp-org:phone:cams" which is located at "http://www.upnp.org/schemas/phone/cams-v2.xsd".

5.3.4.2 Description of fields in the CallBackAvailability structure

```
<?xml version="1.0" encoding="utf-8"?>
<cams:callBackAvailability
  xsi:schemaLocation="urn:schemas-upnp-org:phone:cams
  http://www.upnp.org/schemas/phone/cams-v2.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:cams="urn:schemas-upnp-org:phone:cams">
  <callBack>
    <callBackID>ID of the Call back</callBackID>
  </callBack>
  <!-- Any other Call back (if any) go here.-->
</cams:callbackAvailability>
```

<xml>

Required for all XML documents. Case Sensitive.

<CallBackAvailability>

Required. Root element.

<CallBack>

Required. Contains the list of Call backs that are available and uniquely identified by the CallBackIDs. It includes the CallBackID sub element.

<CallBackID>

Required, Indicates the ID of the Call back.

5.3.5 PushInfo

The format of the PushInfo state variable is an XML document. It includes information that a TS received from a service provider on WAN side or some status changes that occurs within a TS.

ISO/IEC 29341-26-10:2017(E)

5.3.5.1 XML Schema Definition

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for PushInfo in the XML namespace "urn:schemas-upnp-org:phone:cams" which is located at "http://www.upnp.org/schemas/phone/cams-v2.xsd".

5.3.5.2 Description of fields in the PushInfo structure

```
<?xml version="1.0" encoding="utf-8"?>
<cams:pushInfo
  xsi:schemaLocation="urn:schemas-upnp-org:phone:cams
http://www.upnp.org/schemas/phone/cams-v2.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:cams="urn:schemas-upnp-org:phone:cams">
  <priority>priority of information</priority>
  <summary>summary of information</summary>
  <url>URL to get detail information</url>
  <import format="format">
    imported information goes here
  </import>
</cams:pushInfo>
```

<xml>

Required for all XML documents. Case Sensitive.

<pushInfo>

Required. Root element.

<priority>

Required. xsd:string, Indicates the priority of the event. Table 7 shows the allowed values of the <priority> element.

Table 7 — Allowed values for priority

Value	Description
<u>"High"</u>	Indicates high priority event.
<u>"Normal"</u>	Indicates normal priority event.
<u>"Low"</u>	Indicates low priority event.
<u>TBD</u>	<i>(Specified by UPnP vendors.)</i>

<summary>

Required. xsd:string, Indicates the summary of the event.

<url>

Required. xsd:url, Indicates the URL to get detail information.

<import>

Allowed. xsd:string, includes information from the other source. The format of this information is represented by the "format" attribute.

5.3.6 VoiceMailInfo

This state variable is evented by the Telephony Server (TS) when a new voice mail becomes available.

5.3.6.1 XML Schema Definition

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for voiceMailInfo in the XML namespace "urn:schemas-upnp-org:phone:cams" which is located at "http://www.upnp.org/schemas/phone/cams-v2.xsd".

5.3.6.2 Description of fields in the **VoiceMailInfo** structure

```
<?xml version="1.0" encoding="utf-8"?>
<cams:voiceMailInfo
  xsi:schemaLocation="urn:schemas-upnp-org:phone:cams
    http://www.upnp.org/schemas/phone/cams-v2.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:cams="urn:schemas-upnp-org:phone:cams">
  <voiceMail>
    <voiceMailID>ID of the voice mail</voiceMailID>
    <callee>the callee who left the voice mail</callee>
    <url>URL to retrieve the voice mail</url>
    <duration> duration of the voice mail</duration>
    <time>date, time of the voicemail when left</time>
    <!-- Any other envelope information (if any) go here.-->
  </voiceMail>
</cams:voiceMailInfo>

<xml>
```

Required for all XML documents. Case Sensitive.

<voicemailInfo>

Required. Root element.

<voiceMail>

Required. Contains the following sub elements.

<voiceMailID>

Required. xsd:string, Indicates the ID of the voice mail.

<callee>

Allowed. xsd:string, Indicates the tel uri of the callee who left the voice mail.

<url>

Allowed. xsd:url, Indicates the url to retrieve the voice mail.

<duration>

Allowed. xsd:string, Indicates the duration of the voice mail.

<time>

Allowed. xsd:string, Indicates the time of the voice mail when left in ISO 8601 format.

5.3.7 **ParallelCallInfo**

The format of the **ParallelCallInfo** state variable is an XML document. It includes configuration information of the remote device.

5.3.7.1 XML Schema Definition

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for **ParallelCallInfo** in the XML namespace "urn:schemas-upnp-org:phone:cams" which is located at "http://www.upnp.org/schemas/phone/cams-v2.xsd".

5.3.7.2 Description of fields in the **ParallelCallInfo** structure

```
<?xml version="1.0" encoding="utf-8"?>
<cams:parallelCallInfo
  xsi:schemaLocation="urn:schemas-upnp-org:phone:cams
    http://www.upnp.org/schemas/phone/cams-v2.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:cams="urn:schemas-upnp-org:phone:cams"
  xmlns:peer="urn:schemas-upnp-org:phone:peer">
  <informationType>
    status information of remote device
  </informationType>
```

```

<remoteParty>
  <peer:id>ID of the Peer</peer:id>
</remoteParty>
</cams:parallelCallInfo>

```

```
<xml>
```

Required. Case Sensitive.

```
<parallelCallInfo>
```

Required. Root element.

```
<informationType>
```

Required. xsd:string, Indicates the type of configuration information of a remote TS.

Table 8 — Allowed values for informationType

Value	Description
" ParallelCallRequest "	The TS has received a request for Parallel Call from a remote device.
" ParallelCallAccepted "	The TS has received a response that a remote TS accepted a Parallel Call request.
" ParallelCallDenied "	The TS has received a response that a remote TS denied a Parallel Call request.
TBD	(Specified by UPnP vendors.)

```
<remoteParty>
```

Allowed. peer:peerType, Includes an identity of a remote device which will receive the Parallel Call. If the value of the <informationType> element is "[ParallelCallAccepted](#)", then this element shall contain the valid value.

5.3.8 [A_ARG_TYPE_TelephonyServerIdentity](#)

This state variable contains the unique identifier of a Telephony Server. The identity of a Telephony Server is expressed using the standard URI (Unified Resource Identifier) scheme as specified in [3]. In case of SIP, the identity of the Telephony Server identity contains the SIP URI. In case of a generic resource identified by a telephone number, the Telephony Server identity contains the TEL URI [6].

5.3.9 [A_ARG_TYPE_TelCPName](#)

The [A_ARG_TYPE_TelCPName](#) state variable describes a unique name for a TelCP. Each TelCP has its own name. A TelCP can register its name to the [CallManagement](#) service by invoking the [RegisterTelCPName\(\)](#) action. And a TelCP also can unregister and change its TelCPName by invoking the [RegisterTelCPName\(\)](#) action. It is recommended that this state variable includes human readable and understandable string. (e.g. TV@Living, Alice's TV etc.)

5.3.10 [A_ARG_TYPE_TelCPNameList](#)

The [A_ARG_TYPE_TelCPNameList](#) state variable includes the names of all the TelCPs. The format of the [A_ARG_TYPE_TelCPNameList](#) is comma separated values of sting. The structure of this state variable is the same as the format of the [TelCPNameList](#) state variable.

5.3.11 [A_ARG_TYPE_Expires](#)

The [A_ARG_TYPE_Expires](#) state variable contains the value of the duration for validity of the TelCPName. The value of this state variable is an integer number in seconds.

5.3.12 [A_ARG_TYPE_MediaCapabilityInfo](#)

This state variable defines an XML document that contains the Media Capability of a Media Session.

5.3.12.1 XML Schema Definition

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for [MediaCapabilityInfo](#) in the XML namespace "urn:schemas-upnp-org:phone:cams" which is located at "http://www.upnp.org/schemas/phone/cams-v2.xsd".

5.3.12.2 Description of fields in the [MediaCapabilityInfo](#) structure

```
<?xml version="1.0" encoding="utf-8"?>
<cams:mediaCapabilityInfo
  xsi:schemaLocation="urn:schemas-upnp-org:phone:cams
    http://www.upnp.org/schemas/phone/cams-v2.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:cams="urn:schemas-upnp-org:phone:cams">
  <mediaMixingCapability></mediaMixingCapability>
  <mediaSharingCapability></mediaSharingCapability>
  <enableMediaMixing> 1 or 0 </enableMediaMixing>
  <shareMedia>Media Session ID</shareMedia>
  <mediaCapability format="format of the Media Capability">
    Media capability
  </mediaCapability>
  <nativeMediaCapability format="format of the Native Media Capability">
    Media capability
  </nativeMediaCapability>
  <contentSharingCapability>
    <protocolInfo>metadata information of the content</protocolInfo>
    <!-- other protocolInfo goes here. -->
    <URI>URI of the content</URI>
    <sharingType>transfer or streaming</sharingType>
  </contentSharingCapability>
</cams:mediaCapabilityInfo>
```

<xml>

Required. Case Sensitive.

<mediaCapabilityInfo>

Required. Shall include a namespace declaration for the [CallManagement](#) service Schema ("urn:schemas-upnp-org:phone:cams"). This namespace defines the following elements and attributes:

<mediaMixingCapability>

Allowed. The existence of this tag indicates whether a TS supports media mixing ability. This tag can only exist in the output argument [SupportedMediaCapabilityInfo](#) of the [GetMediaCapabilities\(\)](#) action. This tag shall not exist in any other input or output arguments of the type [MediaCapabilityInfo](#) used by any other actions.

<mediaSharingCapability>

Allowed. The existence of this tag indicates whether a telephony client support the media sharing feature. This tag can only exist in the output argument [MediaCapabilityInfo](#) of the [GetMediaCapabilities\(\)](#) action. This tag shall not exist in any other input or output arguments of type [A_ARG_TYPE_MediaCapabilityInfo](#) state variable used by any other actions.

<enableMediaMixing>

Allowed. Indicates whether a TS needs to mix media from multiple TCs involved in the Call. This tag can only exist in the argument of type [MediaCapabilityInfo](#) of the [StartMediaTransfer\(\)](#) and [ModifyCall\(\)](#) actions. This tag shall not exist in any other input or output arguments of type [A_ARG_TYPE_MediaCapabilityInfo](#) used by any other actions. A value of "1" means that the TS needs to mix media from multiple Telephony Clients involved in the Call. A value of "0" means that the Telephony Server shall stop mixing media from multiple Telephony Clients involved in the Call.

<shareMedia>

Allowed. Indicates to the TS that the session is for sharing the media of the Media Session identified by the value of the <shareMedia> element. This tag is only valid when used in the argument of type [MediaCapabilityInfo](#) of the [StartMediaTransfer\(\)](#) or [ModifyCall\(\)](#) actions and when creating Media Sessions for sharing media. Please refer to 5.2.1 for the appropriate usage of this tag.

<mediaCapability>

Allowed. Indicates the Media Capabilities for the Call which are handled using TC-Based Media Handling mechanism.

format

Allowed. xsd:string, Indicates the format of Media Capability. Allowed values of this attribute are shown in Table 6.

<nativeMediaCapability>

Allowed. Indicates the Media Capabilities for the Call which are handled using TS-Based Media Handling mechanism.

format

Allowed. xsd:string, Indicates the format of Media Capabilities for this element. Allowed values of this attribute are shown in Table 6.

<contentSharingCapability>

Allowed. Used by a TS to indicate whether it supports the TS-Based Content Sharing mechanism defined in 5.2.6.1 by the [SupportedMediaCapabilityInfo](#) output argument of the [GetMediaCapabilities\(\)](#) action. And used by a TelCP to set up a Content Sharing session using the [StartCall\(\)](#) or the [ModifyCall\(\)](#) action, and trigger the Content Sharing process using the [StartMediaTransfer\(\)](#) action.

<protocolInfo>

Allowed. xsd:string. One or more instances of this element are used by the TS to indicate its Media Capabilities (e.g., transport protocols, codecs etc.) for the TS-Based Content Sharing mechanism. A TelCP uses this element to provide the metadata information of the content for the TS to share to the Remote Party. The format of this string element follows the UPnP AV protocolInfo concept as defined in [12].

<URI>

Allowed. xsd:anyURI. used by a TelCP to provide the URI of the content to be shared, used by the TS to locate and retrieve the content.

<sharingType>

Allowed. xsd:string. used by a TelCP to indicate to the TS if the content should be sent as a file or streamed to the Remote Party when invoking the [StartCall\(\)](#), [ModifyCall\(\)](#) or [StartMediaTransfer\(\)](#) action. The allowed value for this element is "transfer", or "streaming". This element is only a hint to the TS, the TS implementation could anyway choose its supported mechanism.

5.3.13 [A ARG TYPE CalleeID](#)

This state variable contains the unique identifier of a Callee. The format of this state variable is URI or [A ARG TYPE TelCPName](#). URI is used when a TelCP creates a Call to a Remote Party in the WAN side. [A ARG TYPE TelCPName](#) is used when a TelCP creates a Call to a Remote Party in the same LAN side.

5.3.14 [A ARG TYPE CallPriority](#)

This state variable contains the priority of the Call. The allowed values of this state variable are shown in Table 9:

Table 9 — allowedValueList for the [A ARG TYPE CallPriority](#) state variable

Value	R/A ^a
" Normal " (DEFAULT)	R
" Emergency "	A
^a R = required, A = allowed, X = Non-standard.	

where:

- "[Normal](#)" indicates a regular Call.
- "[Emergency](#)" indicates an emergency Call (e.g., Call to the police, Call to the fire station etc.).

5.3.15 A ARG TYPE CallMode

This state variable contains one of the modes for a Call. The allowed values of this state variable are shown in Table 10:

Table 10 — allowedValueList for the A ARG TYPE CallMode state variable

Value	R/A ^a
" <u>PHONE-TelCP</u> "	<u>A</u>
" <u>DP-TelCP</u> "	<u>A</u>
" <u>DP-User</u> "	<u>A</u>
" <u>Non-Monopolize</u> "	<u>R</u>
a <u>R</u> = required, <u>A</u> = allowed, <u>X</u> = Non-standard.	

where:

- "PHONE-TelCP" indicates PHONE-Based Call Monopolization mechanism defined in 5.2.2.
- "DP-TelCP" indicates TelCP-Level Call Monopolization mechanism using the DeviceProtection service. See [15].
- "DP-User" indicates User-Level Call Monopolization mechanism using the DeviceProtection service. See [15].
- "Non-Monopolize" indicates a Non-Monopolization Mode Call.

5.3.16 A ARG TYPE CallID

This state variable uniquely identifies the Call. The CallManagement service assigns a unique ID for a Call when either the StartCall() or InitiateCall() action is invoked or it receives an incoming Call. It is recommended to follow [4] for the format of the CallID.

5.3.17 A ARG TYPE SecretKey

This state variable contains a unique key that is assigned by a TS to the TelCP. The TelCP uses this key to authenticate itself to the TS in order to manage a Monopolization Mode Call. This key is assigned when the TelCP invokes the RegisterTelCPName() action.

5.3.18 A ARG TYPE RejectReason

This state variable contains the reason for rejecting an incoming Call or a modification request for a Call.

5.3.19 A ARG TYPE TCList

This state variable includes a list of TCs information associated with a Call. The list includes the UDN and the Media Session ID for each TC.

5.3.19.1 XML Schema Definition

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for TCList in the XML namespace "urn:schemas-upnp-org:phone:cams" which is located at "http://www.upnp.org/schemas/phone/cams-v2.xsd".

5.3.19.2 Description of fields in the TCList structure

```
<?xml version="1.0" encoding="utf-8"?>
<cams:TCList
  xsi:schemaLocation="urn:schemas-upnp-org:phone:cams
    http://www.upnp.org/schemas/phone/cams-v2.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:cams="urn:schemas-upnp-org:phone:cams">
  <TC>
    <UDN>uuid:UUID</UDN>
    <mediaSessionID>Unique Media Session ID</mediaSessionID>
```

```
</TC>
<!-- Any other TC info (if any) go here.-->
</cams:TCList>
```

<xml>

Required. Case Sensitive.

<TCList>

Required. Shall include the namespace declaration for the [CallManagement](#) service Schema ("urn:schemas-upnp-org:phone:cams"). This namespace "urn:schemas-upnp-org:phone:cams" defines the following elements to includes information about one or more TCs:

<TC>

Allowed. Includes information about a TC. It has two sub elements: <UDN> and <mediaSessionID>.

<UDN>

Allowed. Indicates a unique device name of a TC.

<mediaSessionID>

Required. Indicates a unique ID for a Media Session. See [10] in detail.

5.3.20 [A ARG TYPE CallInfoList](#)

This state variable contains a list of ongoing Calls and their information as defined in the [CallInfo](#) structure.

5.3.20.1 XML Schema Definition

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for [CallInfoList](#) in the XML namespace "urn:schemas-upnp-org:phone:cams" which is located at "http://www.upnp.org/schemas/phone/cams-v2.xsd".

5.3.20.2 Description of fields in the [CallInfoList](#) structure

```
<?xml version="1.0" encoding="utf-8"?>
<cams:callInfoList
  xsi:schemaLocation="urn:schemas-upnp-org:phone:cams
  http://www.upnp.org/schemas/phone/cams-v2.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:cams="urn:schemas-upnp-org:phone:cams"
  xmlns:peer="urn:schemas-upnp-org:phone:peer">
  <callInfo>
    <callID>unique ID for the Call</callID>
    <targetNames type="type of the targetName">
      unique names or IDs for targets
    </targetNames>
    <callStatus reason="reason why Call Status is changed">
      status of the Call
    </callStatus>
    <priority>priority of the Call</priority>
    <remoteParty>
      <peer:id>ID of the Peer<peer:id>
    </remoteParty>
    <TCList>
      <TC>
        <UDN>uuid:UUID</UDN>
        <mediaSessionID>Unique Media Session ID</mediaSessionID>
      </TC>
      <!-- Any other TC (if any) go here.-->
    </TCList>
    <mediaCapability format="format of Media Capability">
      Media capability
    </mediaCapability>
  </callInfo>
</callInfoList>
```

```

    <TelCPNames>unique names for the TelCPs</TelCPNames>
    <callStatus reason="reason why status is changed">
      status of the Call
    </callStatus>
  </callInfo>
  <!-- Any other callInfo (if any) go here.-->
</cams:callInfoList>

```

<xml>

Required. Case Sensitive.

<callInfoList>

Required. Shall include the name space declaration for the complex type <peerType> ("urn:schemas-upnp-org:phone:peer") and the namespace declaration for the [CallManagement](#) service Schema ("urn:schemas-upnp-org:phone:cams"). This element includes one or more <callInfo> elements as described in the [CallInfo](#) structure.

5.3.21 [A ARG TYPE CallLogs](#)

This state variable contains the history of the Calls managed by a TS.

5.3.21.1 XML Schema Definition

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for [CallLogs](#) in the XML namespace "urn:schemas-upnp-org:phone:cams" which is located at "http://www.upnp.org/schemas/phone/cams-v2.xsd".

5.3.21.2 Description of fields in the [CallLogs](#) structure

```

<?xml version="1.0" encoding="utf-8"?>
<cams:callLogs
  xsi:schemaLocation="urn:schemas-upnp-org:phone:cams
http://www.upnp.org/schemas/phone/cams-v2.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:cams="urn:schemas-upnp-org:phone:cams"
  xmlns:peer="urn:schemas-upnp-org:phone:peer">
  <call direction=incoming|outgoing success=true|false>
    <type> type of the Call whether audio, video or data</type>
    <caller>
      <peer:id>id of the Caller</peer:id>
      <peer:name>user friendly name associated with id</peer:name>
    </caller>
    <callee>
      <peer:id>id of the Caller</peer:id>
      <peer:name>user friendly name associated with id</peer:name>
    </callee>
    <startDateTime>time in ISO 8601 format </startDateTime>
    <duration>Call duration in ms</duration>
  </call>
  <!-- Any other Call History (if any) go here.-->
</cams:callLogs>

```

<xml>

Required. Case Sensitive.

<callLogs>

Required. Shall include the name space declaration for the complex type <peerType> ("urn:schemas-upnp-org:phone:peer") and the namespace declaration for the [CallManagement](#) service Schema ("urn:schemas-upnp-org:phone:cams"). This namespace "urn:schemas-upnp-org:phone:cams" defines the following elements and attributes:

<call>

Allowed. Contains information relevant to a Call managed by a TS. The [direction](#) attribute identifies whether it is an incoming or outgoing Call and the [success](#) attribute identifies whether the Call was successful or failed. It contains the following sub elements:

ISO/IEC 29341-26-10:2017(E)

<type>

Required. Identifies the type of the Call such as audio video or data Call.

<caller>

Required. peer:peerType, Contains information related to a Caller. It contains the following sub elements:

<peer:id>

Required. Uniquely identifies a Caller (i.e., telephone number or SIP address etc.)

<peer:name>

Allowed. Contains a user friendly name for the Caller.

<callee>

Required. peer:peerType, Contains information related to a Callee. Contains the following sub elements:

<peer:id>

Required. Uniquely identifies a Callee (i.e., telephone number or SIP address etc.)

<peer:name>

Allowed. Contains a user friendly name for the Callee.

<startTime>

Required. Indicates the starting time of the Call. The format of the element is in ISO 8601 format.

<duration>

Allowed. Indicates the entire duration of the Call in milliseconds.

5.3.22 A ARG TYPE CallBackID

This state variable contains the unique identifier of a Call back registration. A TS generates this unique identifier as a result of the Call back registration request from a TelCP. The TS uses this unique identifier in the event notification message to indicate the status of the Call back registration which includes the availability of the Callee.

5.3.23 A ARG TYPE CallBackInfoList

This state variable contains the list of Call backs that are currently registered with the TS. The structure of the A ARG TYPE CallBackInfoList state variable is a XML Document:

- <callbackinfo> is the root element.

Note that since the value of A ARG TYPE CallBackInfoList is XML, it needs to be escaped (using the normal XML rules: [8] 2.4) before embedding in a SOAP response message.

5.3.23.1 XML Schema Definition

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for CallBackInfoList in the XML namespace "urn:schemas-upnp-org:phone:cams" which is located at "http://www.upnp.org/schemas/phone/cams-v2.xsd".

5.3.23.2 Description of fields in the CallBackInfoList structure

```
<?xml version="1.0" encoding="utf-8"?>
<cams:callbackInfoList
  xsi:schemaLocation="urn:schemas-upnp-org:phone:cams
  http://www.upnp.org/schemas/phone/cams-v2.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:cams="urn:schemas-upnp-org:phone:cams"
  xmlns:peer="urn:schemas-upnp-org:phone:peer">
  <callbackInfo>
    <callbackID>identifies a Call back registration</callbackID>
    <callee>
```

```

    <peer:id>id of the Callee</peer:id>
  </callee>
</callBackInfo>
</cams:callBackInfoList>

```

<xml>

Required. Case Sensitive.

<callBackInfoList>

Required. Shall include the name space declaration for the complex type <peerType> ("urn:schemas-upnp-org:phone:peer") and the namespace declaration for the [CallManagement](#) service Schema ("urn:schemas-upnp-org:phone:cams"). This namespace "urn:schemas-upnp-org:phone:cams" defines the following elements and attributes:

<callBackInfo>

Allowed. Includes a specific entry for the Call back registration including the ID of the Call back that uniquely identifies a specific Call back registration and the ID of the Callee.

<callBackID>

Required. Uniquely identifies a Call back registration within a TS.

<callee>

Required. peer:peerType, Includes a unique identification of the Callee such as phone number or SIP address etc. Contains the following sub elements:

<peer:id>

Required. Uniquely identifies a Callee (i.e., telephone number or SIP address etc.).

5.3.24 [A ARG TYPE PushInfoList](#)

The format of the [PushInfoList](#) state variable is an XML document. It includes a list of [PushInfo](#).

5.3.24.1 XML Schema Definition

This is a string containing an XML fragment. The XML fragment in this argument shall validate against the XML schema for [PushInfoList](#) in the XML namespace "urn:schemas-upnp-org:phone:cams" which is located at "http://www.upnp.org/schemas/phone/cams-v2.xsd".

5.3.24.2 Description of fields in the [PushInfoList](#) structure

```

<?xml version="1.0" encoding="utf-8"?>
<cams:pushInfoList
  xsi:schemaLocation="urn:schemas-upnp-org:phone:cams
http://www.upnp.org/schemas/phone/cams-v2.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:cams="urn:schemas-upnp-org:phone:cams">
  <pushInfo>
    <priority>priority of information</priority>
    <summary>summary of information</summary>
    <url>URL to get detail information</url>
    <import format="format">
      imported information goes here
    </import>
  </pushInfo>
  <!-- Any other pushInfo (if any) go here.-->
</cams:pushInfoList>

```

<xml>

Required for all XML documents. Case Sensitive.

<pushInfoList>

Required. Shall include the namespace declaration for the [CallManagement](#) service Schema ("urn:schemas-upnp-org:phone:cams"). This element includes zero or more <pushInfo> elements as described in the [pushInfo](#) structure.